HELSINKI SCHOOL OF ECONOMICS (HSE)
Department of Management

LEVERAGING A PRODUCT PLATFORM MODEL TO SUPPORT SOFTWARE PRODUCT DEVELOPMENT RELATED MANAGERIAL DECISION-MAKING

Case: Evolvis Euro RSCG

Information Systems Science
Master's thesis
Sami Relander
Fall 2003

Approved by the Head of the Department of Management 20 / 11 2003, and

awarded the grade Hyvä, 70 p.

Tomi Dahlberg Timo Saarinen
KTT KTT

ABSTRACT

Objectives

This study focuses on leveraging an existing conceptual, software engineering based, product platform model to help, facilitate and increase the quality of non-technology oriented managerial decision-making within software product (family) development. Based on prior theoretical work in the areas of both software product platform development and managerial decision-making, the Sääksjärvi (1998) model is identified for use as a foundation for designing and producing a tentative, new, conceptual model to aid decision-making and provide decision-support. The new model is produced by separately analyzing, extending and leveraging each component in the Sääksjärvi (1998) model. Some preliminary notions and tentative ideas are also put forth as to the explicit mathematical notations of the new model.

Methodology

For the empirical part of this thesis, qualitative data is collected, from the selected case company's operating unit Evolvis Euro RSCG, from a variety of sources including both formal and informal interviews, workshops and documentation. The primary methodology is a combination of participation in and observation of the chosen case company's operating unit. The qualitative case company findings are plotted into cross-reference tables, based on three constructs structural firmness, functional convenience and representational delight. These constructs are further decomposed into six metrics. The components are subsequently synthesized on a complete, holistic level.

Findings

The final results from this research project are ambiguous, as the presented model achieves a fair overall mark, on a conceptual level, for aiding and supporting decision-making, but fails to add value on an explicit mathematical level. One concludes that the new, presented model can increase the level of decision quality in product platform based software product development but great care must be used in its application into real business environments. The presented mathematical formulas require additional refinement and further research to achieve a viable and valid status among managers and future researchers.

Keywords

Software product development, managerial decision-making, product platforms

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The aim of this first, introductory chapter is to present a solid basis and a sound foundation for the rest of the thesis. In addition to a brief background presentation, this chapter will outline the purpose and the logic of the research. The research problem will be formulated first on a conceptual level, then narrowing it down to a specific, exact set of research questions. A short and concise listing of some of the past research and literature related to the thesis will also be presented. The methodology used herein will be thoroughly discussed, along with an introduction to the case. This chapter ends with a list of essential terminology the Reader should be familiar with, and finally a more detailed outline of the rest of the study.

1.1 Background

In the ever changing, dynamic world of information technology, business management is constantly under pressure to make decisions, with far reaching effects, on the very many different aspects of software product development. Difficult and often contradictory challenges have to be met, with cost cutting and efficiency concerns on the one extreme and the importance of succeeding in strategic product development on the other. In essence, business management is being forced to do increasingly more with less, under conditions of economical uncertainty, technological change and market unpredictability.

It is often the case that business management does not have sufficient technological understanding or enough technical expertise to be able to make sound, justifiable decisions on the product development issues at hand. Having to rely on subordinates and external specialists can have the detrimental effect of biasing a manager's decision towards technological arguments, missing key strategic business imperatives. Keeping in mind that software product development issues cannot be answered solely from a technological perspective, it is clear that business management is in need of help in aligning and integrating product development with the overall business strategy of their respective companies. This can only be achieved by increasing the understanding of software product development related issues, articulated by justifiable reasoning, expressed with sound decisions and visible through firm actions.

Modern software product development applies many models and methods mainly to help and ease design, decrease development time and maximize quality, including e.g. lifecycle planning, evolutionary prototyping, joint application development and staged delivery (McConnell 1996; 133, 433, 449, 549). The overwhelming majority of these models and methods focus on the processes, implementation or application of software development within its different phases. In addition, the models and methods are generally built from a bottom-up technological perspective with the software developer or systems engineer as the primary target audience.

Until recently, very few models have focused on supporting managerial decision-making in software product development. As stated previously, the main area of attention in software product development has been on tactical level execution, rather than on strategic business integration and alignment. Efficiency concerns on the software project level have outrun the effectiveness issues on the development process level. However, based on recent studies (e.g. McGrath 1995; Meyer & Seliger 1998; Sääksjärvi 1998, 2002) the emergence of product platform models into the software product development domain has been shown to have, not only productivity benefits with decreased development time and increased quality, but also increased strategic direction and business strategy integration.

Despite the advances and progress in software product development within strategic business contexts, it can still be argued that business management is in need of simple tools and applicable methods to help facilitate high quality, generic managerial decision-making in software product development within the boundaries of business strategy, organizational limitations and market requirements. The following study tries to explore and examine these issues, hoping to shed some new light on the matter, leading to new and improved software product development decisions.

1.2 Research problem, questions and objectives

This sub-chapter begins with a short discussion on the research problem by first presenting a broad framework, then a more specific context. Next, a set of research questions is formulated. Finally, the study's objectives will be discussed.

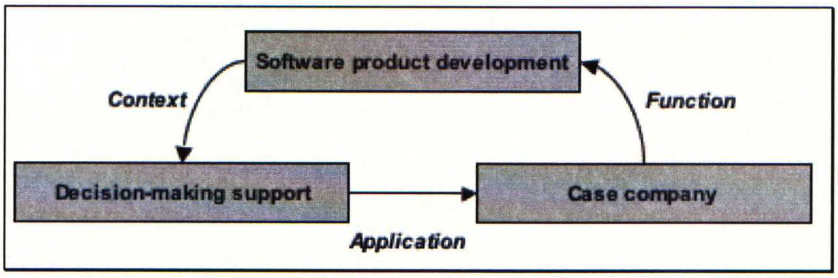
1.2.1 Research problem

Software is the essential element in modern information technology, taking many forms and functions ranging from embedded software to operating systems to business programs to entertainment applications. Among its many tasks, software connects hardware, binds operational logic, links work processes and represents output information.

Using software can have its problems, but they are generally regarded as simple and straightforward when compared with the issues and concerns behind the design and development of complex software systems. Lehman (1998) has even gone so far as to argue that measured in structure, content and functionality software systems are by far the most complex artifacts mankind has ever created. Based on the difficulties behind successful software product development, a significantly large percentage of projects fail to meet requirements and expectations (Sambamurthy & Kirsch 2000, 391-392). Naturally, the reasons contributing to this factor are diverse in character and numerous in number.

The research problem of this thesis, therefore, centers on the notion that current software development practices are not working to the extent they should be. Without examining the exponential amount of possible explanations for this failure, this study focuses on the beginning phases of software product development, by isolating decision-making as the key component under scrutiny. The simple rationale behind this logic is that by increasing the effectiveness of decision-making, i.e. making the correct decisions, software development projects should have a better chance of initial survival and later success. In other words, software development provides a context and framework for supporting decision-making, which in turn is applied to a case company (in the empirical part of this thesis) with a clear software development function. Figure 1 elaborates on this general research problem framework.

Figure 1: Broad research problem framework



Isolating the decision-making and software product development components deepens the level of research problem analysis, from a broad framework to a more specific context. In table 1 changing decision structure[1] is plotted vertically, while horizontally depicting increasing business intensity from an operational stage through a tactical tier to a strategic level. The question, therefore, is if the unstructured nature of product development related decisions could be a reason for software's fluctuating levels of success? This leads to an interesting research problem: could the possibility of succeeding in software product development be increased by restructuring the nature of the underlying problems, from unstructured ones to semi-structured ones? Rephrasing the question from a managerial perspective it simplifies to: can managerial decision-making be supported in software product development? The arrow pointing upwards in table 1 clarifies the research problem.

Table 1: Specific research problem context

	Operational	**Tactical**	**Strategic**
Structured decisions	E.g. inventory control	E.g. variance analysis	E.g. Placement of new plant location
Semi-structured decisions	Decision support systems application area		↑
Unstructured decisions	E.g. designing a magazine cover	E.g. managerial recruitment	E.g. new software product development

Source: Adapted from Turban & Aronson 2001, 12.

1.2.2 Research questions

Building on the previously presented specific research problem context, the level of analysis will be further deepened by introducing a set of practical, concrete research questions. From table 1 it is clear that a tool will be needed to facilitate the move from an unstructured decision to a semi-structured one. It is commonly known that the majority of software product development models and methods are technologically oriented (see e.g. Abrahamsson et al. 2002), and thus not directed at or suited for managers and executives. Therefore, the use of a traditional software development method must be ruled out, in trying to increase the quality of software product development by supporting managerial decision-making. As will be shown and discussed throughout this thesis,

[1] The process of decision-making falls along a continuum ranging from structured (or programmed) to unstructured (or non-programmed) decisions. Structured decision processes involve routine, repetitive underlying problems, which can be solved with standard solutions and methods. Unstructured decision processes, on the other hand, involve complex and fuzzy problems for which no obvious solution exists. Semistructured decision processes fall between the two extremes. (Turban & Aronson 1998, 11-12)

this tool is a software product platform, which will make the transformation possible[2]. This leads to:

Research question 1: *Could a product platform, or its application, be used to help improve the quality of managerial software product development decisions?*

If the answer to research question 1 is positive, one would need to further elaborate on the issue, by deepening the analysis to:

Research question 2: *What are the key new elements needed to apply to a product platform for it to provide managerial decision support?*

If the new elements are clear on a conceptual level, it would be interesting to see if they might be extended into mathematical formulas with:

Research question 3: *Is it possible to transform conceptual product platform leveraging ideas into specific mathematical equations?*

Finally, a comparison of the new, leveraged product platform model to the initial one would be appropriate to close the theoretical cycle. This will be achieved by:

Research question 4: *What are the differences, in form and function, with the new, leveraged model to the initial product platform model?*

1.2.3 Research objectives

The objective of this thesis is, naturally, to answer the research problems presented previously. In addition, it is hoped that the study will provide some basis for future theoretical research on using product platforms to support managerial decision-making. A further wish is that the case company, to which the results of the theoretical part will be applied, could benefit from the results.

1.3 Target group

Although this thesis can provide useful insight into decision support within a software product development context for a variety of people from different backgrounds, it is nevertheless targeted at senior, non-technical business management and executives who seek concepts, ideas and methods to support their software product development related decisions.

To facilitate the comprehension of the ideas and concepts herein, it is a necessity that the Reader has a descent working knowledge on key concepts within software product development, managerial decision-making, finance, micro-economics and business strategy.

[2] The selection and use of a specific product platform application will be discussed in more detail in chapter 3.1.4.

1.4 Previous research

The focus of this sub-chapter is to present some seminal research relating to this study. The presentation is by no means a conclusive and thorough literature review, but will suffice as a brief examination of some of the key, related issues. The research is divided into two parts. First, influential research on managerial decision-making will be presented, followed by decisive studies on product platforms.

1.4.1 Decision-making

The decision sciences are a broad field within the larger spectrum of social sciences, and have seen a steady growth in popularity and applicability over the past several decades. Key focus points of research within decision-making have been e.g. the value of information, utility theory, contextuality of decision-making and various aspects of supporting or increasing the quality of decisions. All of these concepts also relate to the field of information systems science research. The following, brief list of research most relevant for this thesis is presented in chronological order in table 2. The research spans a time period of approximately the past 16 years.

Table 2: Previous research on decision-making

Researchers	Key findings of study	Implications to this thesis
Ahituv & Neumann, 1987	Decisions are based on current knowledge available. New information continuously revises knowledge. → Information is to be evaluated in context with the decisions it seeks to support. Identifies the normative, realistic and subjective value of information.	Ties in the decision-making process with the evaluation of information and application of knowledge.
Allen & Hauptman, 1990	The communication of decisions and information among technical professionals in R&D project teams increases productivity.	Communication of information, sharing of knowledge and articulation of decisions are key important elements when determining R&D success.
Lindberg & Zackrisson, 1991	Presents the element of forecasting in managerial decision-making. Forecasts can be used for predictions, the generation of knowledge, policy guidance and system performance monitoring.	Forecasting, or the ability to predict consequences from a sequence of actions, is a key component in decision-making. Decision-making always has multiple effects.

Researchers	Key findings of study	Implications to this thesis
Singh, 1994	All managerial decisions involve both a qualitative and quantitative aspect. Qualitative information and knowledge needs to reflect and support decisions based upon quantitative facts.	A distinction among strategic and tactical decisions. Stresses the importance of qualitative information with regard to mere quantitative facts.
Teng & Calhoun, 1996	Examines IT decision-making relationships within the emerging organizational computing environment. Explores managers' perceptions as a facilitator of their decision-making activities.	Managers recognize the value of generic non-application specific information technology development environments.
Chu & Spires, 2000	Discusses the importance of reducing complexity in improving decision quality. Evaluates the joint effects of effort and quality on decision strategy choice.	Reducing complexity will increase the quality of decision-making. Incorporating two cited theories; cost-benefit and bounded rationality.
Schmidt, Montoya-Weiss & Massey, 2001	Studies the effectiveness of new product development decisions, using escalation of commitment theory. Findings suggest that teams make more effective decisions than individuals.	Cross-functional teams outperform individuals in development activities. Therefore, the use of IT and decision support systems (in this case groupware) should be used to streamline coordination and to facilitate better process management.
Van Bruggen & Wierenga, 2001	Studies the impact of decision support systems on managerial decision-making. Success of DSS depends on alignment with problem-solving process. For DSS to be successful a real and acute demand must exist	For successful decision support to be possible, a clear problem-solving process must exist. Successful decision support systems are created only based on a specific need.
Verma, Thompson, Moore & Louviere, 2001	Presents an integrated framework for designing profit-maximizing products. Introduces operating difficulty as a function of product and process attributes.	Overcoming operational difficulties is a prerequisite to the successful creation of profit maximizing products and services.

Researchers	Key findings of study	Implications to this thesis
Ranganathan & Sethi, 2002	Presents the concept of rationality in strategic information technology decisions. Draws upon structural and resource-based perspectives of strategy to examine the influence of shared domain knowledge in strategic IT decisions.	Rational and logical decisions must be made for a successful software product to be completed. Rationality and logic cannot be overcome by complexity and uncertainty. Common and shared knowledge of product development can lead to better design and production.
Venkatesh, Speier & Morris, 2002	Presents the approaches to understanding user perception formation about technology. Puts forth an integrated model to understand how user perceptions are formed prior to a system implementation.	To succeed in software product development simplicity is key as this increases and unifies understanding and knowledge in an organization.

Summarizing these articles from the broad field of the decision sciences is not an easy task, as their variety and diversity in scale and scope is huge. Nevertheless, highlighting and showcasing these articles should provide some idea as to the magnitude of the subject. On this note, it is difficult, if not impossible, to provide any kind of valid and reliable synthesis on the matter.

1.4.2 Product platforms and platform concepts

The product platform and skeleton concepts presented herein are by no means new ideas in the product development / application framework research field. Over the past ten or so years, many researches have approached the problems and concepts from different angles ranging from product architecture to production process to paradigm shifts, thus providing a myriad of studies to use. However, the vast majority of studies are technologically orientated providing little value to this thesis. The research originating from a business and strategy context, and therefore most relevant to this thesis, is presented briefly, in chronological order, in table 3. The research spans a time period of approximately the past eleven years.

Table 3: Previous research on product platforms and platform concepts

Researchers	Key findings of study	Implications to this thesis
Wheelwright & Clark, 1992	The definition of the product platform concept in a physical product development context. The distinguishing of the initial platform, its extension, and its renewal.	The formulation of key terms and concepts used in this and other studies.
Meyer & Utterback, 1993	Emphasizes the meaning and importance of the company's personnel's core skills in product platform development. The product platform is seen as an agent for developing and storing the company's strategy and retained skills.	The extension and leveraging of a company's strategy, through the use of an agent, i.e. a platform, to decrease product development, marketing, and production etc. redundancy.
Welke, 1994	Presents a paradigm shift from neo-classical, craft-based, software development to new roles (user-arrangers and product developers) and development strategies (object-oriented platforms and off-the-shelf objects)	Provides some interesting insight into the possible future of software development. Written in 1994, the materialization of Welke's ideas have emerged as e.g. C++ and Enterprise Java Beans technologies.
McGrath, 1995	The modeling of product strategies into four tiers: the vision, product platforms, product lines, and individual products. Distinguishing between a platform strategy and a product line strategy.	The emphasis of technology as the common denominator. The defining technology derives all other concepts and parts of the product platform. One of the seminal source materials for this thesis.
Ullrich & Eppinger, 1995	Organizes products into five categories based on a product development viewpoint. Forms a basis for the use of the "*platform*" term.	The definition of "*platform products*" as being built around and upon a common, technical core environment.
Meyer & Lehnerd, 1997	The product platform strategy can also be used as a tool guiding marketing strategy. A successful product platform requires the right organizational structure and model.	The use of the initial product platform concept within a wider organizational context. Defining changes to product platforms.
Meyer, Terzakian & Utterback, 1997	The product platform is the technological foundation and basis for product families. Some mathematical notations on platform effectiveness and efficiency.	The physical implementation of the product platform's technical design including the architecture and its interfaces, production technology and processes.

Researchers	Key findings of study	Implications to this thesis
Sääksjärvi, 1998	Presents the product platform concept and its three key components product architecture, future extensions and technology strategy.	The leveraging and extension of the Sääksjärvi model to help aid decision support in software product development. Forms the core material for this thesis.
Rajala, 2000	The evolution of the product platform concept to an integrated application framework within a business strategy concept.	The application of product platform thinking into a business concept. Testing product platform thinking in practice with a case company.
Sääksjärvi, 2002[3]	Extending the Sääksjärvi 1998 model to include implementation. Incorporates product architecture with the manufacturing process. Introduces sub-strategies and their overall fit.	An extension of the Sääksjärvi 1998 model into a more practical realm, but still remaining in a software design and business strategy context.

Based on the presented previous research, three clear, distinct, and linear evolution paths can be identified. Firstly, it is obvious that product platform thinking began with physical products and mechanical engineering, and only with the gradual development of information technology, has it moved into the field of information centric products i.e. software.

Secondly, research has tended to be from a technology or engineering sciences background and only recently has a shift into a business context been witnessed. Finally, there is a clear change from emphasizing the development of a product platform to the application and alignment of one in a business context. Two key drivers of change, namely the development and evolution of IT in general and the convergence of engineering and business sciences, have in part, facilitated these three evolution paths.

1.5 Methodology

According to Soininen (1995, 75-90) and Hirsjärvi et al. (1997, 122-123) there are three basic, traditional research strategy types: exploratory research, quantitative survey and case study. The three types are not mutually exclusive and can co-exist within a single study. To some extent, the three types also complement, conflict and contradict each other. Based on this classification, it is easy to determine that the theoretical part of this thesis can be categorized as exploratory research,

[3] When comparing the 1998 and 2002 product platform articles of Sääksjärvi, it is clear that the emphasis in the first one is on the effectiveness of product platforms i.e. "*doing the right thing*". In the second article the emphasis clearly shifts to the efficiency of product platforms i.e. "*doing the thing right*".

whereas the empirical part as a case study. Hirsjärvi et al. (1997, 127-128) provide a second schema for organizing research based on the goals of a study. Therefore, research is either exploratory, explanatory, descriptive or predictive. Hence, this thesis can be described as exploratory for its theoretical part and descriptive for its empirical part.

Based on the exploratory nature of the theoretical part of this thesis, no specific hypotheses can be presented for testing. The goal can therefore only be in presenting new ideas, uncovering hidden assumptions, arranging shortcuts for communication and portraying the complexity of the underlying problem. The new, leveraged model that will be presented can help in the identification of aggregated entities and the specification of mutual, interconnected relations among the elements under study. (Hirsjärvi et al. 1997; 131, 135, 148)

Moving on to the empirical part of this thesis, it is often stated that the primary approach of a qualitative study is to describe and express real life situations, in this case, a company operating within a business environment. Qualitative studies usually aim at finding and uncovering facts, rather than validating or testing existing propositions and hypotheses. Characterizing qualitative research is usually done along its direction (e.g. naturalist), methodological foundations (e.g. phenomenology) or research approaches (e.g. grounded theory). However, common to all types of qualitative research is their holistic nature towards information gathering. In addition, the sample is always collected in a real environment, in a real situation, stressing the importance of a human collector. Qualitative research tends to favor inductive analysis when investigating a suitable, convenient sample. Finally, the research plan can and should adapt to changing circumstances during the actual, unique study. (Hirsjärvi et al. 1997, 152-155)

Based on the nature of the to be presented new, leveraged model in the theoretical part, it is easy to see that the use of quantitative methods would not, within the context and limitations of this thesis, be appropriate. Therefore, a qualitative case analysis and examination of the leveraged model in a single business context must suffice. A second important reason for choosing a qualitative examination is to seek initial and preliminary reflections, thoughts and comments on the new, tentative model. This thesis is not looking for absolute answers, exact numbers or definitive comparisons, as the model to be presented is only in its infancy and does not, at its current stage in development, require or need these.

1.5.1 Overview of case study research

Yin (2003, 13) defines a case study as "*an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident*". Bourdreau et al. (2001, 4) classify a case study as involving a concentrated assessment and inspection of a small number of entities. In case studies independent variables are not manipulated nor are confounding variables controlled. One of the major concerns with case studies is to generate knowledge of the particular, from which analytic generalization[4] is possible (Bourdreau et al. 2001, 4).

Yin (2003, 1) states that case studies are the preferred research strategy and method when "*how*" or "*why*" questions are being asked, when the researcher has little control over events, and when attention is directed at a contemporary phenomenon within a real life context. By intensively studying a small number of entities, a case researcher should develop a thorough understanding of a phenomenon, from which hypotheses may be generated (Boudreau et al. 2001, 4). A case study must also cope with a technically distinctive situation where there are many more variables of interest than can be collected. This is due to the fact that evidence is generally collected from multiple sources. (Yin 2003, 13-14)

The method used for this case analysis will be a subset of action research, which aims at both taking action and creating knowledge or theory about that action. Contrary to traditional research, which foremost strives for knowledge creation, the result from action research is both a practical action and a research product (Coghlan & Brannick 2001, xi). Being participative by nature, action research is best suited for research situations where a series of events and actions over time in an organization need to be investigated to improve aspects of it (Coghlan & Brannick 2001, xi-xii).

The subset of action research this thesis uses is called "*participation in and observation of*" the case company[5]. In its simplest form this type of research is best described as having an informal research design with the researcher being one of the subjects under examination (Hirsjärvi et al. 1997, 181). With observation the researcher can find out if the subjects under examination are acting the way they say they will (Hirsjärvi et al. 1997, 199). Participation with

[4] Analytic generalization, also known as theoretical elaboration, is the study of a phenomenon in a specific set of conditions to support or elaborate a theory or model (Boudreau et al. 2001, 4).
[5] For more information on "participation in and observation of" case study research please refer to e.g. Eskola & Suoranta 2001, 98-99, 126-130; Uusitalo 1991, 89-90; Yin 2003, 92-93.

the subjects, on the other hand, is suited for when the researcher is e.g. an employee, like the subjects (Hirsjärvi et al. 1997, 203).

1.5.2 Data collection procedures

In this sub-chapter, the methods and data collection procedures are presented, described and explained, to allow the Reader to assess and evaluate the feasibility, applicability, validity and reliability of this study (Hirsjärvi et al. 1997, 241-242). As earlier stated, the research method for the empirical, case study part of this thesis was conducted based on a subset of action research, namely participation in and observation of the selected case company (e.g. Eskola & Suoranta 2001, 98-99, 126-130; Uusitalo 1991, 89-90; Yin 2003, 92-93). The generic *"participation in"* part has been an ongoing process from the inception of Evolvis (the selected case company, presented briefly in chapter 1.6), whereas the specific, scientific *"observation of"* part has been conducted during a six-month period ranging from May 2002 to the beginning of November 2002. All formal[6] interviews were conducted during May 2003 – August 2003 as half-structured, confidential theme interviews. All of the interviews were recorded and transcribed.

The researcher has had access to an overwhelming range of material, diverse both in scale and scope. This material includes and consists of memos, minutes, various other documents and documentation, project and client information, and software code. All of the findings and evidence presented herein have been gathered and collected from various sources including physical artifacts, official interviews[7] as well as intangible, unofficial discussions and conversations. It is to be noted, that while the overwhelming majority of findings are based on scientifically qualifying evidence, a small percentage is, however, based on subjective, tacit interpretations and thus do not qualify as justifiable, valid and reliable proof. They have, nonetheless, been presented because of their tight integration with real evidence and the fact that removing tacit knowledge from clearly articulated information would be, at best, highly time-consuming, if not totally impossible.

Hirsjärvi et al. (1997, 208) state that in qualitative research, especially in case studies, information is being gathered concurrently from various sources and with different methods. Therefore, the analysis of the gathered information cannot be done in a distinct phase, but rather is

---

[6] A set of informal interviews was conducted during January and February 2003. These interviews were not recorded and transcribed.

[7] Please see appendices A and B for more detailed information relating to the interviews and interviewees.

being done simultaneously with the collection of information. This has also been the procedure with the case study of this research.

1.5.3 Case study research questions

The formulation of the case study research questions is based around the answer to the simple question whether the tentative, leveraged model, presented in the theoretical part of this thesis, could be applied into a real business context decision-making situation. Building on this rather vague question, a set of more specific and practical questions is introduced based on three constructs presented in the Information Systems Research September 2002 article "*Businesses as Buildings: Metrics for the Architectural Quality of Internet Businesses*" by Kim et al. From the three constructs[8] six metrics are adapted, each of which transforms into precise case company research questions[9].

The selected constructs are structural firmness, functional convenience and representational delight, and will be adopted into this thesis as a basis and synthesis for relevant empirical case study research question generation and application. These three, high-level, constructs decompose into two practical metrics each, from which specific, relevant questions can be derived. The construct-metric-question axis forms an abstract-practical polarization that provides background, depth, context and meaning to the questions. Beside this use, the constructs are not relevant for this study, and thus will not be analyzed further.

Of the three constructs structural firmness refers to the solidity of system structure, both from an internal and external perspective, thus having two metrics, internal stability and external applicability (adapted from Kim et al. 2002, 239-242). Functional convenience is used for collecting information and processing decisions (adapted from Kim et al. 2002, 239-243). Finally, the representational delight construct defines the interface aspects between the model and its users in an organizational context. In addition, this construct evaluates the model's ability to communicate with other entities, either human or e.g. other models. For additional clarity, all of the above-mentioned constructs, metrics and research questions are presented in table 4. (Adapted from Kim et al. 2002, 239-243)

---

[8] The three constructs are originally based on a model used to analyze buildings. The original model dates back to the Roman ages. Kim et al. refer to the work of Gideon 1941 and Rasmussen 1959.

[9] Kim et al. 2002 use sophisticated statistical methods to refine their metrics. Due to the limitations and restrictions of this thesis, a mere qualitative analysis must suffice.

Table 4: Case company analysis constructs, metrics and research questions

Construct	Metric	Research questions
Structural firmness	Internal stability	What is the level of internal fit between the three components of the model? Do they function in a coherent and consistent way?
	External applicability	Is it possible and viable to apply the new leveraged model to the case company and its decision-making processes? What is the organizational fit of the model?
Functional convenience	Information type	What type of data or information does the new model need to produce relevant results? What is the relation of the cost of acquiring this information to the benefit from the results?
	Decision processing	How does the new, leveraged model align in with the decision process? Can it be used on a generic level to support decision-making or is it restricted to a specific phase in decision-making?
Representational delight	System interface	How does the new leveraged model communicate and integrate with the decision-making process of the case company?
	Communication interface	How does the new, leveraged model communicate and interact with external entities? On what level does the communication work best? Graphical, text, or numerical?

Source: Adapted from Kim et al. 2002, 239-246.

1.6 Introduction to the case

In the empirical part of this thesis the leading Finnish communications and PR consultancy, BNL Euro RSCG Oy and in particular one of its operating units, Evolvis Euro RSCG, will be examined. Although Evolvis cannot be described as a pure software product company, it will however provide an insightful basis of analysis for evaluating the theoretical findings of this thesis.

The business strategy of Evolvis is to support and extend the traditional communications and PR activities conducted by its parent company BNL. This is achieved by providing professional consultancy services integrated with a set of software applications to produce solutions that leverage communications that support the overall business objectives of the client.

The software applications Evolvis uses are not sold separately to clients, nor are they ever solutions or ends by themselves, but rather means to a communications and PR end.

Currently Evolvis only employs twelve professionals, a third of which can be categorized as being technical personnel. The researcher is one of the founders of Evolvis and its former head. Currently he is working in Evolvis as a part-time senior consultant and project director Founded during spring 2001, Evolvis has a steadily, organically growing profitable business, which is not blinded or disturbed by IPO-dreams, venture capital fantasies or rapid-growth illusions, as has been witnessed with several Finnish and international companies over the past few years.

1.7 Study scope and limitations

For this study to effectively and efficiently reach the goals and objectives presented earlier, certain restrictions and limitations must be imposed. Therefore, software production processes, tactical design issues, operational implementation concerns or other technical problems will not be examined herein. These interests are generally the predominant focus areas in software engineering.

Secondly, throughout this thesis several models and frameworks will be utilized. Some will act as building blocks while others will be altered to suit the requirements of the study's theoretical part. In doing so, no explicit or implicit value judgment, criticism, disrespect or disapproval is intended towards the models or their authors.

Finally, this thesis is not limited to or confined by any one specific business sector or industry. Thus, it is equally suited for software vendors, systems integrators, IT-consultancies as well as internal corporate IT-departments. The Reader should also note that this thesis does not present a final solution to the addressed issues and problems. Rather, it should be read as a contribution to an ongoing dialogue and continuum of development in the combined and integrated fields of software product development and managerial decision-making.

1.8 Essential terminology

To minimize the Reader's possibility for confusion in this study, it is of great importance to clearly define all of the key terms and concepts used herein. Although the meanings' of some of the terms listed below may seem obvious to the Reader, they have nevertheless been presented. Seminal terms and concepts are also discussed in their relevant chapters. The terms that follow are presented in alphabetical order, and have been divided into two separate tables, to highlight the importance of the theoretical terms.

Table 5: Essential theoretical terminology

Term	Brief description
Product development	A process or sequence of actions designed to facilitate a formal structure for designing, planning, producing, and implementing a product for its intended end use.
Product family	"*A set of individual products that share common technology and address a related set of market applications*" (Meyer & Lehnerd 1997, 35).
Product platform (skeleton[10]*) model*	"*A product architecture and platform concept aiming to increase the reuse of the underlying technology and prolonging product platform derived product life cycles*" (Sääksjärvi 2002, 2). "*A set of common components, modules, or parts from which a stream of derivative products can be efficiently created and launched*" (Meyer & Lehnerd 1997, 7).

Table 6: Other key terms and concepts

Term	Brief description
Indifference curve	A collection or set of compromises between two goods (X and Y) that are equally desirable. Indifference curves always slope downward, fill the plane, never cross and are convex. Holds utility constant. (Landsburg 1999, 57)
Open source software	Software that is available with its source code. Does not necessarily mean free software but freedom to alter and modify source code. Largely centered around the Linux development community.
Unified Modeling Language (UML)	A descriptive definition and specification system that is used to depict system entities, their properties and relations. Allows all project participants to view a system in a consistent and coherent way.
Utility	A numerical value to measure relative ranking of a specific amount of pleasure or satisfaction. Utility is mainly used in micro economics.

[10] Sääksjärvi introduces the term "*skeleton*" to distinguish a platform's use in software engineering and the use of the "*platform*" term in ordinary language. For the purposes of this study the differences are not relevant and both terms will be used interchangeably.

1.9 Study outline

On a broad level, this thesis is structured into four distinct parts. After the first introductory chapter, which has outlined the foundations for this thesis, the basics of managerial decision-making and software product development will be presented in chapter 2. Together, these two chapters, i.e. part I, form the introduction and background to this thesis.

The main theoretical portion will be presented in part II in chapters 3 and 4, where chapter 3 presents the Sääksjärvi (1998) product skeleton model. In chapter 4 the Sääksjärvi model will be leveraged to produce a tentative, new, conceptual model.

Chapters 5 and 6 make up the empirical, third part of the thesis, by first presenting the case company in chapter 5 and then the results, recommendations and discussions in chapter 6. The final chapter summarizes the key findings and concludes with some closing thoughts on the thesis. This thesis ends with a list of references and three appendices. Together the final chapter, the references and appendices form the fourth and final part of this thesis. All of the above-presented information is summarized in table 7.

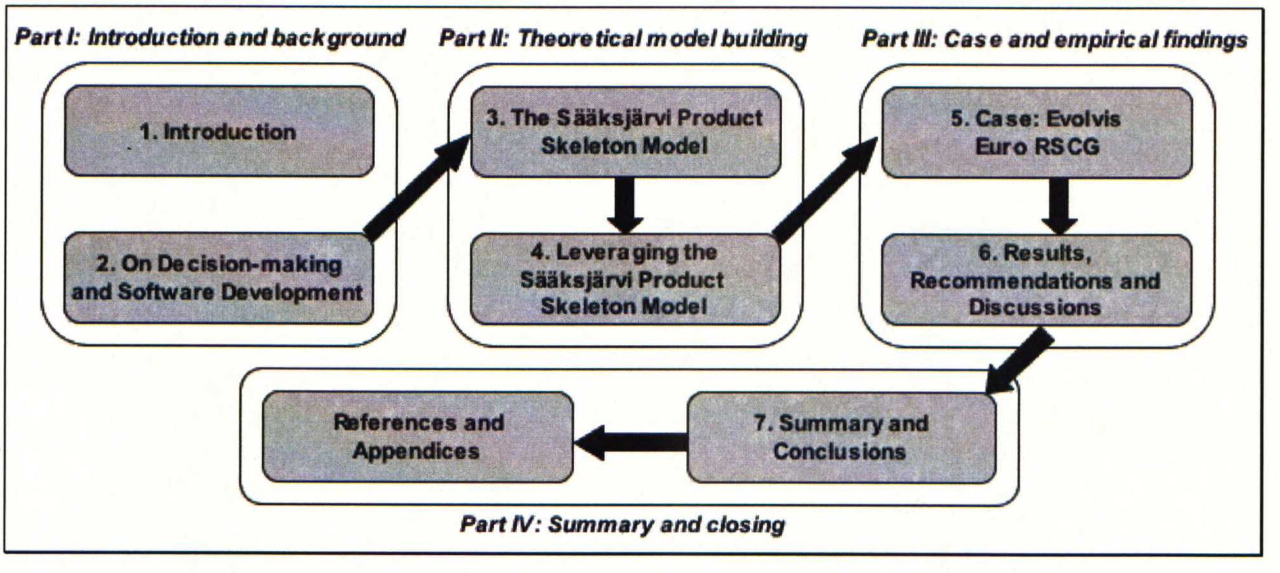
Table 7: Key points of study outline

Part	Chapter	Nature	Objective
I	1. *Introduction*	Not available.	To present the basic concepts, goals and methods used in this study.
	2. *On Decision-making and Software Development*	Positivist, descriptive.	To provide an overview of the two key areas analyzed herein.
II	3. *The Sääksjärvi Product Skeleton Model*	Positivist, descriptive.	To overview product platforms in general and present one specific application, the Sääksjärvi product skeleton model.
	4. *Leveraging the Sääksjärvi Product Skeleton Model*	Exploratory, normative.	To attempt the design of a tentative, new, leveraged model based on the Sääksjärvi product skeleton model.
III	5. *Case: Evolvis Euro RSCG*	Descriptive.	To describe the case and then to apply the tentative, new, leveraged model to it.

	6. *Results, Recommendations and Discussions*	Descriptive, normative.	To summarize the results from the application of the tentative, new model and present some practical recommendations for the case company. To seek further understanding of the deficiencies associated with this thesis.
IV	7. *Summary and Conclusions*	Not available.	To review this thesis and end with some closing thoughts.
	References and appendices	Not available.	To list the literature used in this thesis and to present the appendices.

For additional clarity, the study's outline is pictured in figure 2. The information corresponds with that of table 7.

Figure 2: Study outline in a graphical format



2. ON DECISION-MAKING AND SOFTWARE DEVELOPMENT

This second chapter deals with two focus areas, namely managerial decision-making and software product development. Although acknowledging the impossibility of presenting both subjects thoroughly, the basic and fundamental ideas and concepts will be briefly discussed to allow a smooth transition into the main theoretical part of this thesis. It is also noted that managerial decision-making is a much broader subject for analysis than software product development. Therefore, the emphasis is on scale, when presenting managerial decision-making and scope, when presenting software product development.

2.1 Fundamentals of managerial decision-making

For an organization to succeed in a competitive market, managers and executives must have the ability to make correct decisions at the right time, under the constraint of insufficient, imperfect information. No longer can these individuals rely on hunches, intuition or various habits to provide them with the required insight to successfully tackle the myriad of options, unknown consequences and probabilistic events that continue to occur in their respective organizations. (Baird 1989, xi)

According to Baird (1989, xii) good decision makers, instead of viewing a decision as a single one-off event, possess the ability to think in terms of results, outcomes, probabilities and entire strategies. They don't confuse opinions with facts, but rather analyze advantages and disadvantages, separate fact from fiction and finally select the most appropriate option. Baird (1989, 4) goes on to state that bad decision makers choose a course of action and then seek argumentation to justify and validate it.

The successful implementation, the art and science, of decision-making under complex, uncertain situations often requires the decomposition of the problem into smaller parts. This approach helps in the systematic determination of appropriate goals, assists in comparing available alternatives and aids in estimating probabilities. Therefore, the ultimate goal of this process is to increase the quality of decisions, not to provide absolute, definitive answers. (Baird 1989, 5-6)

In the following sub-chapters, some background and foundations to understand managers' decision-making will be presented, including the characteristics of the types of decisions this study is interested in and decision-making processes. Finally, some aspects on computerized decision-support systems will be discussed.

2.1.1 Decision situations and characteristics

For the study and analysis of decisions to be relevant and viable, certain conditions and characteristics must be met. It would not be worthwhile studying decisions that are unimportant, recurring, simple or happen with certainty. Along these lines, Baird (1989, 6) makes the categorization that decisions must be important or major to justify further, more in-depth analysis. Therefore, minor or even irrelevant decisions where the consequences of a mistake are trivial are not worth the effort of research. Second, a decision must be unique, implying that recurring decisions can be modeled, programmed i.e. solved and then delegated. (Baird 1989, 6)

Third on Baird's list is the notion that decisions worth analyzing must allow some time for study. This means that split-second decisions that must be made immediately are not of interest for the purposes of this study. Fourth, decisions must be complex i.e. involve more than one decision maker, have multiple objectives and contain numerous variables. Finally, uncertainty has to be closely associated with the decisions to be examined, as actions that occur with certainty are not interesting. (Baird 1989, 6)

Baird's above-presented list is relevant for achieving the goals of this study. The problem this study is looking into is always of key importance while simultaneously being very unique and company specific. Due to the scale and scope of the problem, product platform based software product development will inevitably allow for time and resources to be used in its solving process. Hence, the decision to solve the problem can, and should, never be made hastily. Lastly, complexity and uncertainty are key attributes affecting the decision-making regarding the problem this study seeks to solve.

Another important aspect to note about decisions can be made based on the outcomes of the chosen courses of action. There are three possible categories: certainty, risk and uncertainty. Of these three, certainty is the easiest to define, simply meaning that something happens with a 100 % probability i.e. it *will* happen. On the other extreme, with uncertainty the probability with which something will happen is not known. Risk lies somewhere along this continuum, in between the two extreme alternatives. For example, casting a die has a one in six chance of landing on any number i.e. it happens with a known probability between zero and 100, and is therefore categorized as risk. (Baird 1989, 6-8)

Based along the notions of categorizing probabilities into certainty, risk and uncertainty, it is clear that solving the issues regarding product platform based software product development can be seen to lie somewhere between complete uncertainty and expected risk. Accumulating experiences with product platforms could partly be responsible for dissolving the clouds of uncertainty and mystery surrounding successful software product development. On the other hand, breaking the concept of a product platform up into smaller components could produce explicit probability functions, leading to the materialization of risk over uncertainty.

2.1.2 Decision-making processes

Because of the complexity, vagueness and ambiguity of decision-making processes, researchers and scholars have not been able to agree upon a generic, sequential set of steps that would be found in each process. For example Turban & Aronson (2001, 39-57) provide only a very lightweight categorization of a decision-making process into an intelligence, design and choice phase. On the other hand, Baird (1989, 10-14) only reluctantly identifies the stages that are needed as a minimum. First, the problem must be defined, as without a problem no decision can exist. Next one must list all the possible options, i.e. at least two courses of action must be available to the decision maker. The third step would be to define the criteria upon which the problem and its outcome can be evaluated. The final two steps would be to analyze the options by using various methods of computation, estimation and comparison to provide the optimal alternative and then choosing that alternative.

For this thesis and its case analysis, the exact definition and categorization of what constitutes a decision-making process is not compulsory. The key element this study builds upon is the concept that decision-making is a process, a sequence of steps, and a linear course of action. Therefore, it is not necessary to go into the academic debate and discussion concerning the proper and accurate definition of a decision-making process.

2.1.3 Computerized decision support systems

In its simplest form, a system is merely a collection of resources, concepts and procedures used to perform an identifiable function or goal (Turban & Aronson 2001, 34). This form can materialize in simple concepts, depicted as graphical models. However, with the increasing complexity of problems, witnessed with vast amounts of variables, time series and differing interpretations, the need to use computerized support for decision-making has become evident. These systems and computer programs are referred to as decision support systems or DSS.

Early definitions (Turban & Aronson 2001, 96; Hess et al. 2000, 2) of a DSS identified it as a system intended to support managerial decision-making in semi-structured problems and as a system used to supplement and extend a manager's capabilities but not replace his or her judgment. This definition suits the needs of this study well as it seeks to support managerial decision-making by increasing decision quality, and not by providing exact answers.

In distinct contrast to the passive use, clerical user and mechanical efficiency goals of the electronic data processing era of the 1960's and 1970's, DSS gradually began to emerge during the early 1980's to support active use, a management user and effectiveness goals. Today, decision support systems are an integral part of corporate culture in medium and large sized firms throughout the world. DSS can be as simple as an Excel worksheet containing a few functions and macros or as complex as a custom built ERP-integrated legacy system. (Turban & Aronson 2001, 94-115)

A decision support system is typically composed of four subsystems (Turban & Aronson 2001, 100-101). A database management system (DBMS) stores relevant data, a model base management system (MBMS) contains the quantitative models and methods used to analyze the data and a knowledge-based management system (KBMS) can provide intelligence to augment the decision maker's own reasoning. Finally, a DSS has a user interface subsystem for input and output communication. From the perspective of this thesis, the MBMS subsystem is most relevant, as it is essentially for this the new, leveraged model is being designed and produced for.

As to the future of DSS, Hess et al. (2000) argue that intelligent software agents and increasingly specific applications will lead the evolution in the years to come. Advances in interoperability and enhancements in content are being implemented through Open Database Connectivity (ODBC), data warehousing and web-enabled data access. Despite this apparent progress, the MBMS is considered the least developed component and remains the focus of current research in DSS. (Hess et al. 2000, 1-2)

2.2 Fundamentals of software product development

Although perhaps the most recognized and publicized aspect of software development, programming, is not the only component needed for success in developing software products. With equally important phases such as specifications requirements, planning, design, testing, documentation and support, the emphasis should shift to managing software product development on a holistic level. One of the major problems associated with successfully implementing software product development is the fact that it is simultaneously a generic and unique process. On the one hand, the main phases, key issues and major problems relating to a successful software development are generally known, while on the other hand each development initiative is a unique, one-of-a-kind project.

In addition to the previously presented concerns, intense and rapid technological change can at best become manageable, at worst be overwhelming. Starting a software product development project with a specific, new technology is no guarantee for its relevance or usefulness once the project finishes. If this were not enough, clients are constantly demanding increased software quality and shorter development times with fewer financial and human resources.

This sub-chapter explores just some of the very many issues behind software product development, beginning with a set of selected development strategies and then moving on to more practical development tactics, and ending with some concrete development operations.

2.2.1 Development strategies

A classic example of a product design strategy is the distinction between a client / market pull strategy or a technology push strategy. The client pull strategy is based on increasing client requirements that pull the development of a product. A technology push strategy is evident from technology's continuing progress and advance, creating a pulling environment for all its associated products and services.

Michael Earl (1989, 67-94) proposes a more detailed description for possible design directions, articulating, the now famous, top-down, bottom-up, and inside-out tracks. Of the three, the top-down alternative is centered on business objectives and goals that are analyzed based on critical success factors, which in turn provide the requirements specifications for successful software product development. Essentially, the top-down way of arranging a design strategy works by management dictating needs to lower levels of the organization.

The bottom-up alternative is based on evaluating current products and their functionality by users and specialists. Here, surveys and audits are primarily used for gathering information. Therefore, the critical questions are asking what could be done with the existing applications and how could they be enhanced. The third and final direction is created by opportunities and possibilities enabled by information technology in general. This alternative can be initiated by anyone identifying a creative or innovative chance in techniques, technologies or processes. Hence, its name inside-out. (Earl 1989, 67-94)

2.2.2 Development tactics

Commercially successful software development is a complex and uncertain process, which has been documented in many previous studies (see e.g. Lederer & Prasad 1992, Gibbs 1994, Jones 1994, Lehman 1998). To deal with these issues and concerns, researchers, software engineers, consultants and programmers have, over the course of several decades, developed numerous procedures, varying practices and different methods – so-called traditional software development methods – including e.g. the renowned waterfall lifecycle model (McConnell 1996, 136-139). Despite a varying degree of success with the implementation and use of these traditional methods, successful software development is often still a largely elusive, even unreachable goal.

With the emergence of a rapidly expanding Internet and telecommunications software industry including e.g. entertainment, corporate and learning applications, together with new development environments and technologies, have resulted in an increasing need to partly or completely redesign earlier development practices and methods (Abrahamsson et al. 2002, 3). Recently Nandhakumar and Avison (1999) and Truex et al. (2000) have argued that traditional methods are no longer up to the task of supporting and steering software development, but have rather become fictitious, symbolic or hypothetical entities presenting and providing an image of control or normative guidance, with little or no relevance and meaning to actual software development.

The above-presented challenges have, over the past few years, led to new and interesting software development methods being presented by corporate practitioners and business consultants. The emphasis has been on simple, light, fast, flexible and nimble methods, avoiding complex, mechanistic and scientific practices. These methods, broadly categorized under the umbrella of "*agile methods*", have provided a mixture of recycled old components, fresh concepts and interesting ideas (Abrahamsson et al. 2002, 19; Cockburn 2002, xxii). Table 8 briefly presents six major development methods; three traditional methods and three agile methods.

Table 8: Six major development methods

Broad categorization	Development method	Description
Traditional methods	Waterfall Lifecycle	The most well known development method. Document driven. Six phases that do not overlap. Works best with a stable product definition, when working with well understood technologies, and with complex projects. Provides project structure to support diverse teams. (McConnell 1996, 136-139)
	Spiral Lifecycle	A risk-oriented lifecycle model that breaks up a software project into miniprojects. Each miniproject addresses one or several major risks until all have been addressed. Then the model continues execution as a waterfall lifecycle model. (McConnell 1996, 141)
	Code-and-fix	A common development model. If another model isn't explicitly chosen, a code-and-fix model is typically used by default. Starts with a general idea of what to develop then a combination of coding, debugging and testing is applied resulting in a possible release. Has two advantages: little overhead and requires little expertise. (McConnell 1996, 140)
Agile methods	Extreme Programming	Initiated from the problems associated with long development cycles in traditional methods. Consists of five phases. Key principles include short releases, simple design, continuous testing, pair programming, collective ownership and on-site customer. (Abrahamsson et al. 2002, 18-24)
	Scrum	Developed to manage the development process, does not require a specific development practice. Consists of three phases. Key principles include flexibility, adaptability and productivity. (Abrahamsson et al. 2002, 29-34)

Broad categorization	Development method	Description
	Rational Unified Process	Developed to complement and enhance UML, a common modeling method. Only suited for object-oriented development. Consists of four phases. Key principles include iterative development, requirements management, component based architectures and quality verification. (Abrahamsson et al. 2002, 55-59)

2.2.3 Development on an operational level

Finally, to round off this discussion on software development, some operational measures used in practice will be presented. These measures and issues can be roughly divided into two parts, the environment and programming languages used. These two subjects make up the everyday operational concerns associated with developing software.

The environment is perhaps the most important operational decision that has to be made. It has far-reaching effects and thus cannot be taken lightly. Some will argue that the choice of an environment is a tactical level decision or even a strategic decision, but from the point-of-view of this thesis it is operational, because it essentially has a technical answer, derived from purely technological arguments. The environment not only covers the development environment and tools used but also more fundamentally the environment the software is being designed and built for.

Today there are two major environments for development. Microsoft's .Net architecture and the J2EE consortium controlled by Sun Microsystems. Of these the .Net architecture is a *programming language independent environment*, meaning that any number of programming languages can be used to design and produce software meeting the .Net requirements. The most common languages in use are, however, C#[11] and VB.Net, both Microsoft products. .Net software runs exclusively on the Windows operating systems. The other major environment is J2EE, also known as Java 2 Enterprise Edition, which is *an operating system independent environment*. Essentially, this means that software written in Java will run on any operating system, as expressed in Sun's Java slogan "*Write once, run everywhere*". Because of limitations in scale and restrictions in scope, the differences and similarities between the .Net architecture and J2EE cannot be discussed further.

The second major choice that has to be made is that of the programming language. This decision is closely related to the choice for an environment. With the .Net architecture there is

[11] Pronounced "*C-sharp*", C# is a scaled-down version of C++, much like Java.

considerably more freedom in choosing an appropriate programming language, but C# and VB.Net are the only real choices, as the environment is still in its infancy[12]. The natural and only choice in J2EE is the Java programming language.

[12] The current version of the .Net architecture, as of June 2003, is 1.1.

3. THE SÄÄKSJÄRVI PRODUCT SKELETON MODEL

Here, in chapter three, some of the fundamental issues relating to product platforms will be discussed. Then, one specific product platform, the Sääksjärvi (1998) product skeleton, and its individual components will be examined. The chapter continues with a discussion on the implications of the presented model. Finally, a preliminary applicability test is conducted as a basis for the work reported in chapter 4.

3.1 Overview of product platforms

The concepts of product development and product platforms have their background within physical products and mechanical engineering, originally being developed and conceived to improve and increase the effectiveness of product development. The primary focus of mechanical engineering has often been on the production process of a (physical) product rather than on the underlying fundamental architecture and its components.

With an increasingly growing importance in modern society and industry on information-centric or digital products, i.e. software, the main focus can no longer be on the production process, as the pressing of CD-ROMs or the distribution of these products via the Internet, play a very insignificant role when determining total costs. Therefore, the emphasis in software product development platforms is, not only, very much on the underlying architecture and structure of the platform itself, but also on the relations and interfaces between and among the individual components that make up the platform. (Sääksjärvi 1998; 5, 14)

The product platform, as a conceptual framework, is directly linked and related to the concept of reuse, as an efficiency-increasing asset. The product platform is the core of a modular architecture. Together with the technological framework the product platform forms the basis for a product generating system. The product platform should not be seen as merely a passive design concerning technological architecture but also as an active development philosophy integrating many different aspects and views on product development. The core promise that a product platform holds is their possible use as a market-leveraging tool, when developing product variant based global software aimed at a very narrow niche market. (Sääksjärvi 1998; 2, 4-5, 14)

The Sääksjärvi (1998) model focuses on the planning and design phases of platform creation[13]. The analysis stresses the importance and significance of product architecture and technology strategy requirements specification in designing a platform that allows for the generation of a complete set or family of products, without altering the underlying platform. (Sääksjärvi 1998; 2, 4-5)

Sääksjärvi (1998, 14-15) identifies three levels for analyzing the meanings and roles of software product architectures. Firstly, the business sector level, which implies the growing need to adhere architecture models with international standards, thus ensuring compatibility. Secondly, on the level of an individual software company, where the increasing complexity of products leads to consider the use of external partners and sub-contractors. Thus, the architecture plays an important role in determining the form of the organization and the skills needed to manage it and product platform development.

Thirdly and finally, on the software and its development level, the architecture acts to reduce complexity, as perceived by various constituencies, allowing for an accurate depiction of the platform in all of its development phases. These roles and levels of meaning can also be argued to increase the importance of product platform thinking as an agent of competition, ultimately acquiring a de facto standard for itself in product development strategy. (Sääksjärvi 1998, 14-15)

3.1.1 Product platform definitions

Originally a product platform (model) has been defined as follows: "*A product architecture and platform concept aiming to increase the reuse of the underlying technology and prolonging product platform derived product life cycles*" (Sääksjärvi 2002, 2). Although this definition is still valid and useful, providing a more in-depth analysis of the term to deepen the understanding of the concept and its underlying mechanisms should prove constructive. The product platform concept will also be defined from the point-of-view of what it is not to provide context and an overall picture.

McGrath (1995, 45-46) identifies five key characteristics needed to define the concept of a product platform. Firstly, product generations are defined by product platform life cycles and not by individual products. Just as products, platforms too have life-cycles, having distinct beginnings and ends. Secondly, platforms can be extended by introducing incremental periodic improvements.

[13] Later work by Sääksjärvi (e.g. 2002) focus more on the implementation and application of product platforms.

As a third point, McGrath argues that platforms can be used to create other derivative platforms. Fourthly product platforms are essential, not incremental, meaning that a new product introduction has incremental revenue effects when compared with the introduction of a new platform. Finally McGrath identifies that platforms take significantly longer to develop than individual products based on an existing architecture.

Sääksjärvi (1998, 18-19) also presents a set of definitions to specify the product platform concept. His first argument is that the goal of a product platform is to maximize the effectiveness of product development (product R&D) by maximizing the relation of platform derived product sales to the total R&D costs of platform and product development. The second point Sääksjärvi makes is that a platform is a common pre-implemented core used to derive future products and other, new platforms. Thirdly, a platform is based on a chosen technology solution that is found throughout the platform and the products that are derived from it.

Fourthly, the products derived from the platform are placed within a reasonable distance from the original market segments. Sääksjärvi concludes his definitions with the fact that the success of a product platform can only be evaluated after its implementation and use. Thus, an abstract plan or design of a platform cannot be a successful platform. A final point worth noting is that a product platform is never a product by itself and is thus never marketed or sold separately. (Sääksjärvi 1998, 18-19)

Continuing the analysis on platform definitions, an understanding of both the effectiveness (doing the right thing) and efficiency (doing the thing right) of product platforms is needed. Researchers Meyer, Terzakian and Utterback (1997) and Sääksjärvi (1998, 18) simply define platform effectiveness as a ratio of platform derived product sales to product and platform development costs. Platform efficiency, on the other hand, is defined as a ratio of (R&D) costs of derivative products to (R&D) costs of platform version. The answer lies in determining how much the product cost to develop as a fraction of what was allocated to the base platform architecture. By writing[14]: P = platform version, C = costs, S = sales, and $P_n^{prod,n}$ = platform derived products, one is able to formalize mathematical formulas for both platform effectiveness and efficiency.

[14] The same logic and structure for mathematical notations will be used throughout this study.

Table 9: Platform effectiveness and efficiency definitions

Variable	Definition	Mathematical notation
Platform effectiveness	$\frac{\text{Product sales}}{\text{Development cost}}$	$Effectiveness(P_n) = \frac{\Sigma_{P_1}^{P_n} S(P_n^{prod,n})}{\Sigma_{P_1}^{P_n} C(P_n)}$ (1)
Platform efficiency	$\frac{\text{Derivative product costs}}{\text{Platform version costs}}$	$Efficiency(P_n) = \frac{\Sigma_{P_1}^{P_n} C(P_n^{prod,n})}{C(P_n)}$ (2)

Calculating platform effectiveness and efficiency leads back to the core definition of product platforms, as stated by Meyer & Lehnerd (1997, 39): "*...a set of subsystems and interfaces that form a common structure from which a stream of derivative products can be efficiently and effectively developed and produced.*" These subsystems and interfaces essentially form and build the definition of product architecture, one of the core concepts used by e.g. McGrath and Sääksjärvi. When discussing product platforms, it is not enough to state that every product has an architecture. Rather, the goal and focus should be on designing and implementing a *common architecture* across many products, where any single product architecture has the potential to become the basis for this common architecture. (Meyer & Lehnerd 1997, 38-40)

3.1.2 Product platform benefits and risks

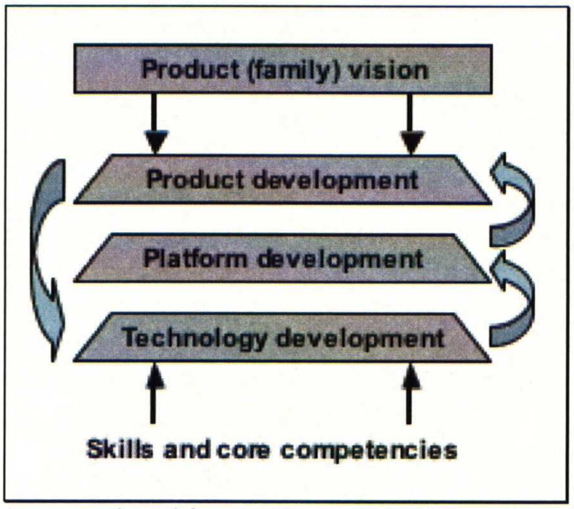
As the foundations and origins of product platforms and platform thinking in general lie among physical products, it is only natural that benefits and risks be evaluated based on this background and context. With physical products, it is often mentioned that the reproduction or copying of the common core or platform is the single most important benefit and asset derived from platforms, resulting in increased development efficiency. A second major benefit, almost constantly presented by related researchers is the point of cumulating corporate knowledge, skills and expertise into the platform, helping the platform developer to align development management with strategic objectives. (Sääksjärvi 1998, 22)

The biggest risk associated with product platforms is the often extensive initial investment that must be made into platform development. On the one side product platforms can have enormous potential for its initial developer by leveraging a common core that can be used to either enter new markets or extend existing ones. A drawback to this potential is the possibility of failure, where the platform becomes virtually useless and thus worthless, essentially becoming a sunk cost that can no longer be recovered.

3.1.3 Technology, product and platform development

Finally, to conclude the synthesis on product platforms, an understanding of the differences between technology development, product development and platform development is required. Although differing from one another, each concept has a distinct function and role. Technology development aims at bringing a specific technology to a point where its application is justified and feasible. The focus does not center solely around technology but also on the skills and core competencies needed for the introduction and management of applicable technologies. Technology, product and platform development relations are presented in figure 3. (McGrath 1995, 44-45)

Figure 3: Technology, product and platform development relations



Source: Adapted from McGrath 1995, 44-46

A major distinction must be made between platform and product development, as they are considerably different in character and nature. The goal of platform development is to create pieces and elements that *enable* the development of products. Products have end-value, platforms have value from the materialization of derived products. These fundamental differences lead to varying investment and evaluation criteria. (McGrath 1995, 44-45)

The driving agent guiding and providing a vector of direction to product development is the initial product or product family vision. Without a strategic vision, there is confusion, misinterpretations and chaos about where to go, what to do, how to do and when to do.

3.1.4 Selecting the Sääksjärvi product skeleton model

For the purposes of this thesis, the Sääksjärvi (1998) skeleton model has been chosen. There are several reasons for this, including the fact that the model studies software product development in a strategic business context, albeit that the components of the model are designed around a technological perspective. Although the Sääksjärvi product skeleton model (1998) is originally defined as a strategic model for product development management, this thesis takes a slightly different view by positioning the model more as a tactical level implementation model.

Secondly, the model is designed as a *platform* model, implicitly implying the possibility to use it as a foundation for future research and extensions. The underlying nature and conceptual foundations of platforms leads to the very essence of this thesis, namely using building blocks on top of a stable basis, leading to the creation of something new. Thirdly, the model is made up of components and their internal relations and interfaces, allowing the model to be broken down to study its individual components. Finally, some interesting work has already been done on the model (e.g. Rajala 2000, Sääksjärvi 2002) that leads to suggest the applicability of the model for future research.

3.2 Model components

Next, each component from the Sääksjärvi (1998) product skeleton model, product architecture, future extensions and technology strategy, will be examined. The components are presented in no particular order.

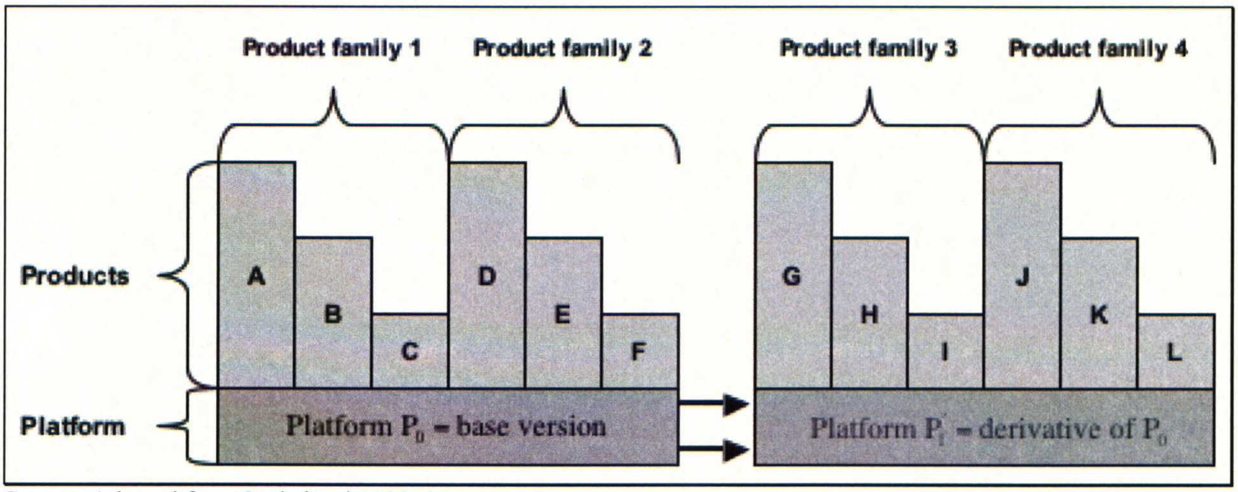
3.2.1 Product architecture

As software products are commonly far more complex than physical products, it can be reasoned that software product architectures are also more complex than physical product architectures. Starting the analysis with the first of three Sääksjärvi model components, product architecture and its elements, interfaces and relations will be examined.

In conjunction with the views of McGrath (1995, 13-18), the product architecture component essentially has three elements, the platform (or skeleton), individual products and product families. The fundamental core of the architecture is the platform, which forms the building block for the other elements. Without the platform, one cannot have a product architecture, as the situation is essentially the same as developing individual products. In addition to acting as the basis for products and product families, the platform can be developed into other, enhanced, platform derivatives.

The second element in product architecture is the set of products built on top of the platform. These products can be categorized in two different ways either by functionality (e.g. professional vs. standard version) and / or end-user segmentation (e.g. corporate vs. home users). A collection or set of products forms the third element in the component namely the product family. Figure 4 depicts all of the component's elements.

Figure 4: Product architecture



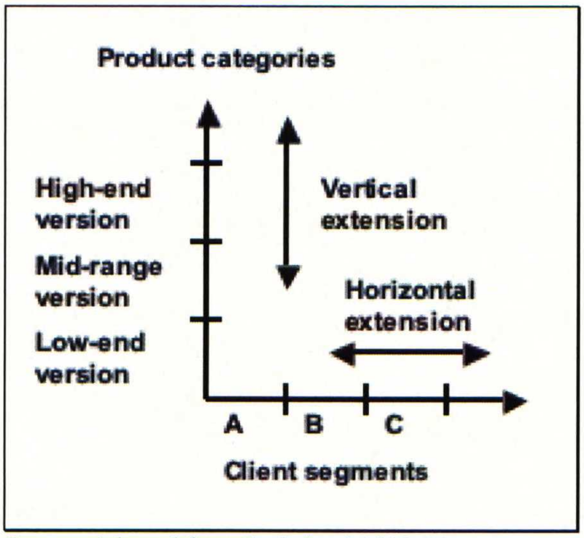
Source: Adapted from Sääksjärvi 1998, 35

3.2.2 Future extensions

Based on the prior research conducted by e.g. Meyer & Lehnerd (1997) and McGrath (1995), Sääksjärvi identifies the future extensions or extension strategy component of the product platform model. Viewing the platform concept from a sales and marketing perspective one could argue its possible use as a tool for market expansion. (Sääksjärvi 1998, 11-12)

Expanding one's market through product platform thinking can be achieved in two ways. Either expand in client segments and / or product categories. The client segment expansion strategy is portrayed on the horizontal axis and the product category's expansion strategy on the vertical axis in figure 5. (Sääksjärvi 1998, 11-12)

Figure 5: Future extensions



Source: Adapted from Sääksjärvi 1998, 12

Client segment extension is achieved when the current end-user[15] segment is extended to incorporate another segment or sets of segments. This extension can pose great risks if the platform cannot be utilized to its maximum in the new segments. (Sääksjärvi 1998, 13)

On the other hand, extensions to the platform can be achieved in product categories by either starting with the high-end version of the product and scaling down or with a low-end version of the product and scaling up. It is argued that the most effective product platform strategies combine and integrate elements from both the vertical and horizontal extension alternatives. (Sääksjärvi 1998, 13)

3.2.3 Technology strategy

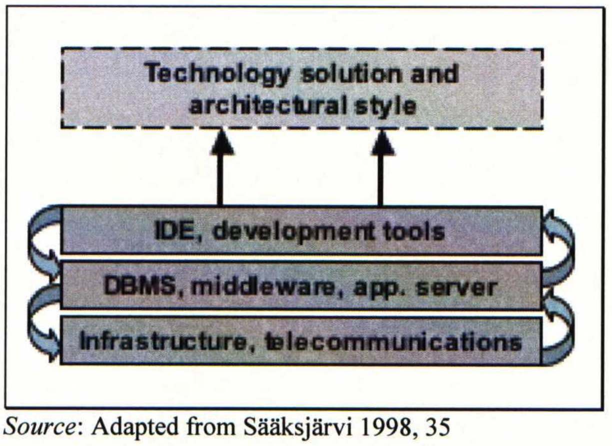
The third and final component of the Sääksjärvi (1998) model is the technology strategy of the platform. Just as the product architecture component consists of different elements, so too does the technology strategy component. The main element of the technology strategy centers around the idea or vision of a predominant technological solution and architectural style. A practical example of this would be, for example, the use of Windows 2000 Active Directory as a basis for a client-server application. Here the use of Windows 2000 Active Directory would be considered the

[15] The extension strategy has to focus on the end-user of the product. Thus, expanding one's reselling channels would not count as an extension strategy. This would be classified as a distribution expansion, which isn't within the limits of this study. Therefore, further elaboration is not possible.

foundation for a technological solution, whereas a client-server implementation would be considered the architectural style.

To support and facilitate the use of a technological solution and architectural style, an additional set of elements is still needed. This set includes the basic infrastructure, protocols and telecommunications needed for the technology strategy. In addition, operating systems, database management systems, middleware and application servers are required. Finally, an integrated development environment with compatible development tools and methods must be implemented for the technology strategy to function in a reliable and consistent way.

Figure 6: Technology strategy



Source: Adapted from Sääksjärvi 1998, 35

McGrath (1995, 42-44) argues that platforms are defined by their most important underlying characteristics, i.e. technology used. While several technologies may be necessary for the creation of a platform, the *defining technology* is most critical, as it establishes the performance characteristics of products based on platforms. McGrath continues that a defining technology provides the primary basis for differentiation but also the limits and boundaries of the product's capabilities.

3.3 Model implications

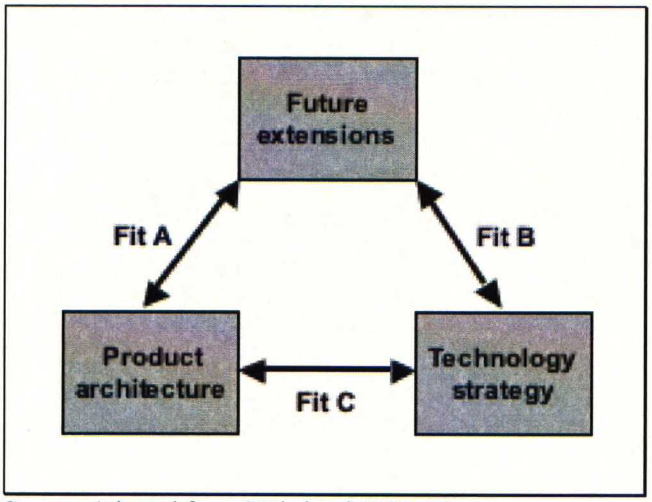
Having presented the Sääksjärvi 1998 model, a further look into the fit between the individual components and the meaning of strategic platform thinking is needed. The fit of individual components will first be examined followed by a discussion on the meaning of strategic platform thinking.

3.3.1 Fit of individual components

An analysis of the Sääksjärvi model can be conducted on two major levels. The obvious, macro-level, is to view the model as a complete, holistic entity providing a vector of direction for solutions to platform and product related issues and concerns. This external perspective gives an overall picture of how the model functions i.e. what it does. An alternative analysis perspective would be to view the model from the inside to understand how it operates. Thus, an understanding of the model's concepts and their relations and interfaces becomes prevailing.

Picturing the Sääksjärvi 1998 product skeleton model, in figure 7, the three individual components and their fits can be seen. Fit A portrays the integration between future extensions and product architecture. Fit B, on the other hand, shows the intensity of future extensions to technology strategy. Lastly, fit C depicts the correlation between product architecture and technology strategy.

Figure 7: The Sääksjärvi 1998 product platform model



Source: Adapted from Sääksjärvi 1998, 35

For an optimal platform solution, it is key that the model be in balance with itself and not be overly tilted or emphasized towards any one direction or component. However, the components of the model can have varying weights attached to them to highlight relative advantages in a specific area. Any one of the components can act as a basis for platform development but all must be implemented for a platform to conceptually exist. (Sääksjärvi 1998, 34)

3.3.2 Meaning of strategic platform thinking

According to Gawer & Cusumano (2002, 54) there is a fundamental difference between the economics of innovating in products for which there is a market and the economics of innovating in the design of a platform for which there is no market. Essentially, platforms are *enabling technologies*, which channel and facilitate complementary innovation in products. Although Gawer & Cusumano approach the world of product platforms from a macro-level, focusing their analysis more on market and industry specific issues, they nevertheless present some interesting perspectives.

Gawer & Cusumano (2002, 2-3) begin by defining a platform as "*an evolving system made of interdependent pieces that can each be innovated upon*". Although being too vague for the purposes of this study, the definition highlights two fundamental phenomena currently impacting the high-tech industry. First, one is witnessing an increasing *interdependency* of products and services. Thus, the basis for development can no longer be on creating individual, stand-alone products and services. Everything must function together and be compatible. Second, an increasing ability and willingness to *innovate* by more actors in the high-tech world is being witnessed. No longer can everything be developed in-house, as has been the norm thus far. Rather product and platform development should be designed and implemented in a way allowing partners and sub-contractors to participate. (Gawer & Cusumano 2002, 2-3)

Gawer & Cusumano (2002, 3) continue that the interaction of the above-mentioned two phenomena constitutes to the emergence of three distinct but interconnected issues: maintaining platform integrity, platform evolution and market leadership in platform environments. In relation to and accordance with Sääksjärvi's views, Gawer & Cusumano emphasize the fit of individual components and elements in maintaining platform integrity. Second is the concern over industry-wide platform evolution. Who is in charge, which mechanisms are used, who exercises final control etc.? Gawer & Cusumano's macro-level view has a direct analogy with the future extensions component in Sääksjärvi's micro-level model. Finally, market leadership issues, whether technical or market-share based, should result in a balance of power among component providers and the provider of the platform technology.

Perhaps the key lesson to be learned from the various researchers studying and working on product platforms is that platform thinking is not a one-time project, but should rather be seen as a process or even an overall philosophy guiding the strategy of the company. Therefore, platform thinking should not, and cannot, be solely conducted on a tactical engineering level, but must, in addition, be incorporated and processed on a strategic, executive tier. The benefits possible from platform thinking will take years, if not decades, to materialize, adding another dimension to the potential difficulties associated with product platforms.

3.4 Preliminary applicability testing of the model

Over the past few sub-chapters, the basics of software product platforms have been covered. Furthermore, one specific application of a platform, namely the Sääksjärvi (1998) skeleton model has been presented. In this sub-chapter, the objective is to conduct some preliminary applicability testing of the model, in the selected case company. The results from this initial examination will be used as guidelines and principles for the attempt to design and produce a tentative, new, leveraged model in chapter 4.

3.4.1 Test overview and design

A preliminary test and examination of the suitability and applicability of the Sääksjärvi (1998) product skeleton model was conducted to assess the initial state of the model. The test's results, and their interpretations, provide some feedback and insight into the possible evolution directions and development paths for Sääksjärvi's (1998) model.

The test was designed to be completed in about 15 minutes, after an initial 30-minute briefing of the study and its intentions. The test only has ten questions[16] (sixteen if the sub-options are also counted), so as to fit on one A4 sheet of paper. The researcher was present during the test to answer any questions.

The test was not intended to be a rigorous, thorough, and scientific examination of the Sääksjärvi (1998) model. Rather, its intentions were to identify broad categories, generic directions or sketchy areas for improvement, refinement and development. The results, presented and discussed next, form the backbone for the work on the leveraging of the Sääksjärvi (1998) model.

3.4.2 Test results and implications

The test was conducted during the formal interviews in May 2003 – August 2003. The test was divided into three parts: general questions, component questions and open-ended questions. The

[16] Please see appendix C for a complete listing of the questions.

test was designed as a one-dimensional Likert-scale with an interval level of measurement. The scale used had the values 1 = disagree totally, 2 = disagree somewhat, 3 = don't know, cannot answer, 4 = agree somewhat and 5 = agree totally. The questionnaire was only given to 10 of Evolvis's employees as one had retired during spring 2003 and one was on maternity leave. Therefore, an $N = 10$ is not a sufficient enough sample to fulfill the requirements imposed for rigorous scientific experiments. However, it sufficed for the preliminary test. The test results are omitted from this report due to privacy concerns[17].

The answers from the first three questions clearly indicate that the model is best suited for a strategic level of use. The respondents seem to be quite unanimous about the fact that the model is not suited for an everyday tool. As for the answers to the second part of the questionnaire, an increasing amount of speculation, uncertainty and ambiguity emerges. The conclusions that can be made are limited to some general observations and broad interpretations.

First, it seems that product architecture is on a solid foundation but might require some more work with regard to its functionality and representation. These observations are equally applicable for the future extensions component. The answers regarding the technology strategy component lead to suggest the need for a more profound revamp.

The third and final part of the preliminary test proved to be the most rewarding and interesting. This was due to the fact that the respondents were not restricted by a scale, but were allowed freedom to express their ideas and thoughts about the model and its components. Among the most insightful comments and remarks made in question 7 were references to the usability of the final software product, budgeting and cost concerns, personnel skills and resources, integration with other project management solutions and customizability. The items presented as answers to question 8 included remarks about flexibility, compatibility and integration. Finally, answers to question 9 were primarily tilted towards expressing that at its current incarnation the Sääksjärvi (1998) model isn't suited for the needs of Evolvis but once developed and customized to the individual requirements and needs of Evolvis, it might be able to serve as a useful aid. To summarize the questionnaire, the following conclusions are drawn:

- The model is best suited for executives, as a strategic tool. It is not suited for everyday, operational software project management.

[17] The results from the test have been presented to the supervisor of this thesis.

- Product architecture and future extensions require further work to better accommodate the needs of decision support. A higher level of abstraction is needed to represent and integrate these two components. Technology strategy requires a more profound redesign.
- In its current form, the Sääksjärvi (1998) model is not suited for supporting software project development in Evolvis. Additional work is required to fit the model into the decision support requirements of Evolvis.

4. LEVERAGING THE SÄÄKSJÄRVI PRODUCT SKELETON MODEL

Previously, in chapter 3, general concepts on product platforms have been discussed. The focus has been on a specific materialization of a product platform, namely the Sääksjärvi (1998) model. Here, in the fourth chapter, the focus is on leveraging the Sääksjärvi model based on the preliminary applicability test, presented and discussed in sub-chapter 3.4. New drivers, innovative logic, creative views and a fresh philosophy will be introduced to support the leveraged transformation of the three components and the complete model.

4.1 Product architecture

> *"Product architecture is very important for the success of our products. It needs to be firm and stable, while simultaneously adaptable and open."*
>
> \- A programmer
> From an informal interview

Of all the Sääksjärvi model's components product architecture is the most important one, as it lays down the foundations for the other components. Without product architecture, the other components become irrelevant, whereas a successful materialization of a product platform could exist without e.g. the future extensions component.

In the following sub-chapters, modularity is introduced into product architecture as a fundamental concept steering and guiding architectural design and style. With modularity consolidation of architectural design and style is possible while simultaneously increasing the level of abstraction to better accommodate the decision-making needs of business managers and executives. In addition, modularity as a concept hides and mitigates most of the complexities associated with product architecture.

4.1.1 Product modularity

Gawer & Cusumano (2002, 4-5) define a *module* as a unit with powerful structural elements connected to each other, while relatively weakly connected to elements in other units. With varying degrees of connection, gradations of modularity can exist. Gawer & Cusumano go on to argue that the consequences of modular designs go far beyond purely technical characteristics of a product. Design decisions can have a profound effect on the organization of production, as a modular design will allow people to divide up the work into tasks or sets of tasks that are relatively independent of each other. Therefore, modularity can have a strong impact on innovation by facilitating improvements on a modular level, which does not threaten the integrity of the overall system. (Gawer & Cusumano 2002, 4-5)

Building on the work of Karl Ulrich, Joseph Pine defines five types of modularity in his 1993 book "*Mass Customization*", as a basis for product and service customization. According to Pine, modularity is the architectural style and design required for successful customization. Although Pine mainly focuses on physical products and services, his work can be seen as a foundation for later research in product platforms and software development. The five types of modularity defined by Pine (1993, 200-211) are presented in table 10.

Table 10: Five types of modularity by Pine

Type of modularity	Description
Component-sharing modularity	The same component is used in multiple projects. Obvious benefits include the reduction of costs, while allowing for more variety and speedier implementation of projects.
Component-swapping / mixed modularity	The complement of component-sharing modularity. Here different components are paired or mixed with each other to produce other components.
Cut-to-fit modularity	The component is continually variable within preset or practical limits. This is best suited for projects that must be constantly varied to match individual client needs.
Bus or back-plane modularity	A standard structure that can attach a number of different kinds of components.[18]
Sectional modularity	Provides the greatest degree of variety and customization, by allowing the configuration of any number of different types of components in arbitrary ways, as long as each component is connected to another with standard interfaces.[19]

Source: Adapted from Pine 1993, 200-211

[18] A good example of back-plane modularity is the motherboard found in computers that acts as a structure for connectivity and data transfer among the other components.

[19] The classic example of sectional modularity is Lego building blocks with their unique, cylinder-type interfaces. The number of different object combinations that can be built with Lego blocks is only subject to the possible limitations of imagination.

Another interesting presentation on modularity is that of Baldwin & Clark in their 1999 book "*Design Rules – The Power of Modularity*", which is presented in an engineering context as opposed to that of a conceptual context by Pine. Baldwin & Clark use modularity in their discussions on complex systems (e.g. Baldwin & Clark 1999, 63-65), to distinguish, on the one hand, between *interdependence within* and *independence across modules*, where modules are units in a larger system that are structurally independent of one another, but work together. Therefore, the system in which the modules exist must provide a framework, or architecture, to allow both independence of structure and integration of function. On the other hand, Baldwin & Clark argue that a complex system can only be managed by dividing it up into smaller pieces and looking at each one separately. If the complexity of an individual element is still to great, it can be isolated by defining a separate *abstraction*, with a simple *interface*, essentially *hiding* the complexity. (Baldwin & Clark 1999, 63-66)

Baldwin & Clark (1999, 90-91) also identify three basic things modularity does. Firstly, modularity increases the range of manageable complexity, by limiting the scope of interaction between elements. In this way, modularity reduces the amount and range of cycling that occurs in a design or production process. With the increasing number of interconnected steps, the process becomes increasingly difficult to bring to successful completion. The time spent goes up, while success probability and output quality go down. Modularity reduces the range and scope of potential cycles, by reducing the set of allowed interactions.

Second, modularity allows concurrent work on different parts of a large and complex design. The individual elements in a modular structure can all be worked on simultaneously. When the benefits of concurrent processing are added to the reduction in time spent in cycling, the timesavings can be dramatic. Third, modularity accommodates uncertainty, by partitioning design parameters into those that are visible and those that are hidden. From the perspective of other designers, working on the same complex system, hidden parameters are essentially "*black boxes*". (Baldwin & Clark 1999, 90-91)

4.1.2 Modular operators

If a complex system displays all of the characteristics of modularity i.e. has a structural form of a nested hierarchy, is built on units that are highly interconnected in themselves, but largely independent; functions in a coordinated way, and each unit has a well-defined role in the system, Baldwin & Clark argue that its analysis and assessment can be achieved with six simple modular operators. These operators, applied at different stages and in various combinations, can generate all possible evolutionary paths for the structure. Baldwin & Clark continue by stating that these operators are a powerful set of conceptual tools, implicit in the logic of modular designs. The six modular operators are: splitting, substitution, augmentation, exclusion, inverting and porting, which are discussed and analyzed next. (Baldwin & Clark 1999, 123)

Splitting and substitution

As the name implies, splitting essentially divides a single-level design with interdependent parameters into a hierarchical design. After the split, each module will constitute a separate design in its own right, which in turn may be split again. (Baldwin & Clark 1999, 132-134)

Competition among alternative module designs can only take place after splitting facilitates module-level substitution. The possibility of substitution forms the basis for economic competition between alternative modules. Splitting and substitution are thus natural complements. (Baldwin & Clark 1999, 134-135)

Augmentation and exclusion

Augmenting simply means adding a module, whereas excluding means leaving one out. Like the previous pair, splitting and substitution, these two operators are complementary. Systems in which a user can initially choose a set of modules to match needs is called "*configurable*". If subsequent module additions or subtractions are possible, the system is "*reconfigurable*", which can take the form of substitutions, augmentations or exclusions. (Baldwin & Clark 1999, 135-138)

Inversion and porting

Inversion describes the action of taking previously hidden information and "*moving it up*" the design hierarchy so that it is visible to a group of modules. Over time, a good solution to a common problem will inevitably be inverted and thus migrate up the design hierarchy. If a component solves a generic problem, it will be separated from its initial context and distributed for wider use. Through inversion, what was once hidden becomes visible. (Baldwin & Clark 1999, 138-140)

The last of the six operators, porting, takes place when a module is made to function in more than one system, under different sets of design rules. Just as with inversion, porting causes a module to move up the design hierarchy, but it is invisible as the designers or architects of the system need not know that a port has taken place. For example, in software product development one often hears of code being "*ported*" to different operating systems. (Baldwin & Clark 1999, 140-142)

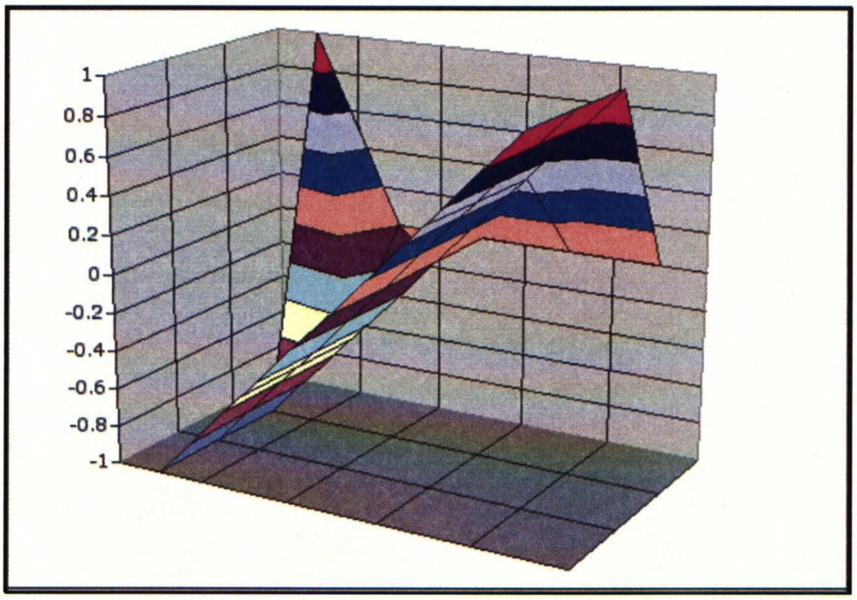
4.1.3 A three dimensional spatial model

Based on the presentation of the six modular operators, it is easy to see that they work naturally as three, complementary pairs, based on their common features and characteristics. Based on this, one is able to construct a three dimensional spatial model by using splitting and substitution as the x-axis, augmentation and exclusion as the y-axis and inversion and porting as the z-axis[20]. The axes are assigned values ranging from -1 to 1, where the extremes represent a pure implementation of either modular operator. A value of zero would represent a combination of the two modular operators.

By using a three dimensional model, incorporating all six modular operators, one is able to achieve two key advantages. Firstly, the model allows for the consolidation of design rules, practices and guidelines into a function or plane, graphically depicted in the model. Second, the graphical format visualizes and emphasizes the key areas and their relative weights, which essentially captures the strategy of product architecture implementations. The obvious disadvantage is the formulation of a relevant function from which the graph could be plotted. The model is presented in figure 8, with a random (e.g. $f(x,y,z)$), computer generated function.

[20] The order of the axis assignment is not relevant and any combination could have been chosen.

Figure 8: 3D spatial model with a random function



4.2 Future extensions

> *"I don't like the future extensions component. It's too different from the others. Why mix marketing with technical issues? I'd like to see it more integrated with the technological core of the skeleton."*
>
> - A project manager
> From a formal interview

Previously, in earlier chapters, the discussion has been focused on what Sääksjärvi sees as the future extensions part of the product platform concept. In this sub-chapter it will be shown that the direction of extensions i.e. vertical or horizontal, is not the relevant question, as is in the Sääksjärvi model. The examination starts with an overview of options (sometimes also referred to as financial options) in general. Then, the analysis is deepened to the concept of real options and its distinct subset, technology options.

After introducing real options analysis (ROA) as a way to value technology investments i.e. software product development projects, and hence use them as managerial decision aids, the presentation comes back to the notion of product vision, covered by both McGrath and Sääksjärvi. Finally, it is shown how a series of technology options are a means to a product vision end. This link between the two concepts is the central concept in the leveraged future extensions component.

4.2.1 Options overview

For many years options have been one of the major research areas in the field of finance. From the early seventies, when Fisher Black and Myron Scholes published their seminal paper "*The pricing of options and other corporate liabilities*", where they present the, now famous, Black-Scholes model, the world of options has seen a steady growth in popularity and applicability with researchers and corporate users alike. Options are used in conjunction with other financial models and concepts to form the building blocks for modern investment theory. (Bodie & Merton 2000, 399-402; Niskanen & Niskanen 2000, 337-340)

Options are part of a larger family of assets called contingent claims, where the future payoff depends on the outcome of some uncertain event. An option can be described as a contract, giving one of two contracting parties the option, but not the obligation, to buy or sell something at an (pre-specified) exercise price at a particular date. Options are thus divided into two categories, call options (buy) and put options (sell). If the option can only be exercised (sold) on a particular date it is called a European call option, whereas if it can also be exercised at any prior date, it is called an American call option. The different types of option contracts e.g. stock, interest-rate and commodity options are traded on international exchanges. (Bodie & Merton 2000, 383-385; Brealey & Myers 2000, 586)

Options[21] have two main applications. Firstly, they allow an investor to modify their risk exposure in relation to the underlying assets, which can be measured by the relation between the option value and the price of the underlying asset. Secondly, options provide an investor, who does not own the underlying asset, with an alternative way to take a position, because the option price is usually only a fraction of the underlying asset's price. (Bodie & Merton 2000, 385-388; Brealey & Myers 2000, 588-590)

4.2.3 Real options

A paradigm shift is currently taking place in investment analysis models. Copeland & Antikarov (2001; 28, 56) argue that over the past 20 years corporations have gradually shifted from the payback method of analyzing investments to the use of net present value (NPV) as the primary model. It is only recently that a shift has begun to move from NPV to real options analysis (ROA),

[21] Within this study, it is not worth going further into calculus examples on options, as many excellent textbooks exist on the subject. Please refer to one of the following books for clarification, elaboration and exercises: Bodie & Merton 2000, chapter 15; Brealey & Myers 2000, chapter 22; Niskanen & Niskanen 2000, chapter 11.

which is defined as a framework for valuing real assets under uncertainty (Erdogmus 2002, 308-309).

A specific investment, project or acquisition can be seen to have a future stream of new investment possibilities related to it. It is even possible that an initial investment with a negative net present value turn positive with successful subsequent related investments. These investment options, called real options (sometimes even called managerial options), are a subset of the options family (Niskanen & Niskanen 2000, 338). Real options allow business executives and managers to increase the utility to their firms by acting to amplify good fortune and by mitigating loss. Many investment opportunities have embedded real options in them, providing management with the opportunity to exercise, if and when it is in the interest of their firm. (Brealey & Myers 2000, 619-622)

Financial literature often divides and categorizes real options into four groups based on the end outcomes from an investment (Niskanen & Niskanen 2000, 338). The first group is identified by the possibility for additional, future, follow-on investments if the initial investment succeeds. An example of this would be, when a firm initially invests in A and after its success and completion invests in B, subject to a constraint where A is a prerequisite to B. The second real option category includes the option to abandon an investment or project, in the case of an unsuccessful initial investment (Niskanen & Niskanen 2000, 338). The business executive or manager has two alternative ways of abandoning an investment, by either selling the investment or asset (this being the obvious choice) or by minimizing future loss on the investment to zero.

The third category in real options is the timing or postponement option, which provides flexibility in the face of uncertainty. Timing options can also provide the opportunity to wait and learn before re-investing (Brealey & Myers 2000, 619). The final category issues the option to vary the firm's output or its production methods, providing flexible production methods. All four option categories are summarized in table 11.

Table 11: Four option categories based on outcomes

Option Category	Description
Follow-on investment	Allows for additional investments, building on the initial investment. The initial investment holds embedded future investment possibilities that can be utilized.
Abandon	Provides the opportunity to abandon the investment either by a sale or by setting future investments on the initial investment to zero.

Option Category	Description
Timing / postponement	During periods of high uncertainty, timing / postponement options allow investment decisions to be moved into the future, i.e. they allow investors to "*buy time*".
Flexible production	An option to exchange one risky asset for another. Allows for flexibility during fluctuations in e.g. demand.

Source: Brealey & Myers 2000, 619-629; Niskanen & Niskanen 2000, 338

To finish off, the key differences between financial options and real options will be examined. Although table 12 is by no means a complete and thorough analysis, it nevertheless provides some useful characteristics to help distinguish the two concepts.

Table 12: Comparison of financial and real options

Variable	Financial options	Real options
Type of market	Complete. A clear market exists for trading.	Incomplete. No distinct market exists for trading.
Type of asset	Traded. Underlying asset is traded on financial markets i.e. stocks and bonds.	Not traded. Underlying asset is very seldom traded. Difficult to trade e.g. projects.
Type of issuance	Issued by third parties i.e. not issued by company they are contingent to.	"*Issued*" by party owning asset as a basis for calculus.
Current price	The current price of the underlying asset can be observed.	The current price of the underlying asset must be estimated.
Discount rate	A discount rate is not needed to value the option.	A discount rate is needed to calculate present value of future payoffs.
Interaction	Self-contained, fixed-structure contracts → no interaction.	Complex and extensive interactions either within a project and / or among projects.
Uncertainty	Only one or two uncertain underlying assets involved → limited uncertainty.	Multiple underlying assets or multiple sources of uncertainty.
Ownership	Clearly defined ownership.	Vague or multiple ownerships.

Source: Adapted from Erdogmus 2002, 346 and Copeland & Antikarov 2001, 111-112

4.2.3.1 Technology options

Numerous studies (e.g. Copeland & Antikarov 2001) argue that for strategic technology centric investments, e.g. a software product platform investment, standard methods such as discounted cash flow analysis, net present value (NPV) and internal rate of return (IRR) do not allow for the element of flexibility to be incorporated, but rather force managers and executives into a fairly rigid form of analysis. Using standard methods therefore typically leads to undervaluation of the investment, forcing managers and executives to rely on subjective intuition to capture the dynamic flexibility of their opportunity. Essentially this means that valuing strategic investments has mainly been a form of art, enhanced with experience, than a scientifically justifiable, objective process.

In contrast, it has been previously discussed that real options can help to identify and recognize key opportunities by facilitating the observation of future events and acquiring information before making any crucial decisions. Restricting real options to the world of technology with its unique characteristics, one is able to define technology options as a distinct subset of real options.

A technology option is unique because it can contain elements from all four option categories presented earlier (Copeland & Antikarov 2001, 121). First it allows for additional investments if the initial one succeeds. Subsequent investments are often necessities when striving to achieve the overall (product) vision. Secondly, as the investment is broken down into a continuous stream of smaller investments, one has the option to abandon the investment after any stage. This produces considerable room for movement by providing flexibility where it is most needed. A strategic investment need no longer be seen as one large investment but as a set of smaller ones connected to each other.

Thirdly, technology options allow managers and executives to "*buy time*" by using the possibility of either postponing or timing a subsequent investment, which can be very useful especially during periods of high uncertainty with regard either to market fluctuations or technological change. Finally, technology options allow for flexible production, as production resources can be moved around fairly freely e.g. between platform development and actual product development.

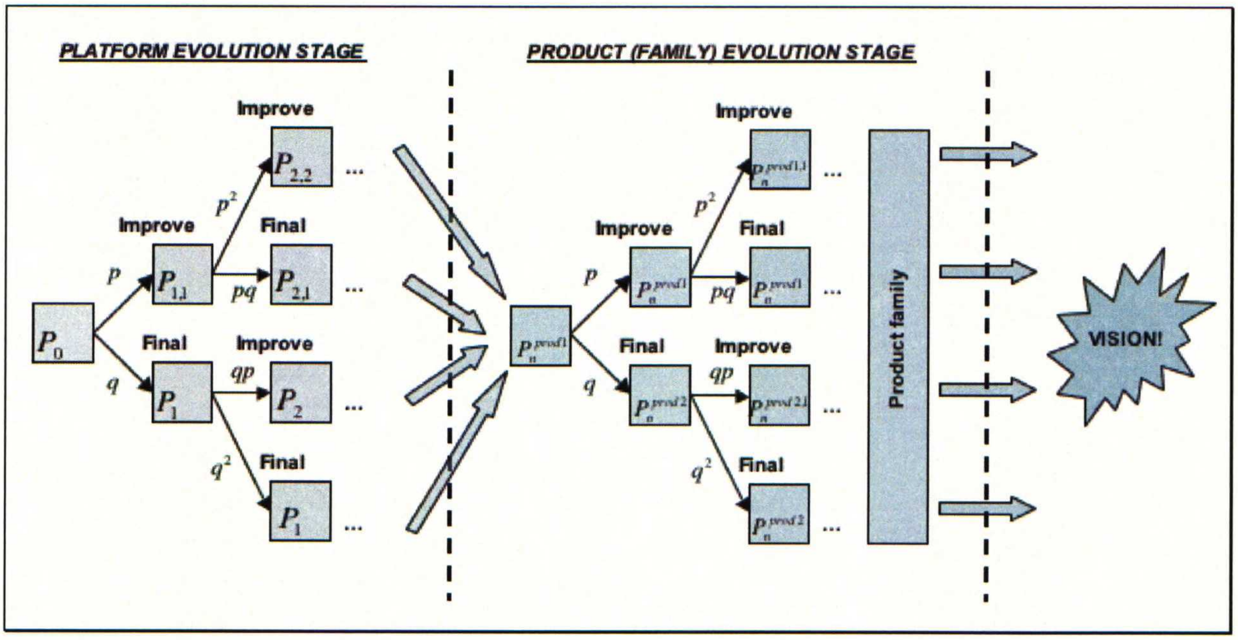
Technology options are thus not just a methodology but also a new way of looking at the decision-making dynamics in technology investments. The two driving elements, providing the benefits for using technology options, are flexibility and the possibility to value this flexibility. Flexibility is a key attribute that can help and facilitate support in the ever-changing technological environments and unpredictable market conditions.

4.2.3.2 A decision tree solution

A decision tree is a simple graphical depiction of a process or sequence of stages from an initial starting point to an end state. Often including probabilities with pay-off calculations, decision trees are commonly used in finance, economics, management science and operations research. A decision tree approach to defining and mapping one's options can often help decision coordination, and therefore also decision quality, during unfolding opportunities.

For the examination of a leveraging solution to the future extensions component, two stages are introduced, making the analysis easier. The component is broken down into a platform evolution stage and a product evolution stage. The final goal and destination with both stages is the product vision, covered equally by McGrath and Sääksjärvi. This presentation begins with the left-hand side in figure 9.

Figure 9: A decision tree solution - from platform to vision



Initially, a base version platform, P_0, is the starting point. Moving along to the right, two possibilities arise; either improve the platform to version $P_{1.1}$ or use the existing version as the final platform, P_1. The improvement alternative takes place with probability p, and the final platform occurs with probability q, where $q = 1 - p$. Note that the version numbering does not imply a specific quality or quantity of development, but rather describes a distinct phase.

The decision here to either improve the platform or make it a final version is essentially an option. With a more detailed definition one could say that it includes the possibility for a follow-on investment, with either a new platform version or a platform based product. Secondly, the option would allow for the abandonment or the underlying asset, i.e. make the platform a final version or abandon development altogether. Third, with incremental, evolutionary development the timing or postponement becomes a viable alternative. Finally, the option facilitates all of the elements needed for flexible production.

With the same logic and structure, one is able to move from the left-hand side to the right, stage by stage. After completing the platform evolution stage, with a final platform version, P_n, the product (family) evolution stage is begun with P_n^{prod1}. In exactly the same way, exercising an option at each decision point, a set of products begins to form, making up the product family. Finally, the product family is the tangible, physical materialization of the vision, which according to McGrath is the strategic goal and desirable end-state of all product development initiatives.

4.3 Technology strategy

> *"Technology strategy is something that needs to be decided by management, based on our skills, talents and interests. Technically oriented people, like me, need boundaries but also require freedom to move within those boundaries."*
>
> \- A programmer
> From a formal interview

To finalize the conceptualization and presentation of the tentative, new model, one needs to look into the technology strategy component and seek a leveraging solution. To facilitate this move and change, two pairs of axes representing four software platform and product characteristics will be presented. Their Edgeworth box equilibrium solution will then be explained.

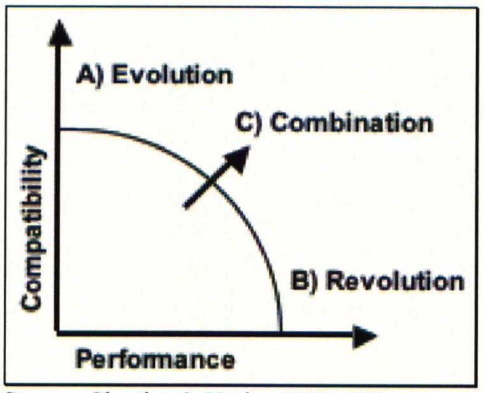
This chapter begins by presenting the performance – compatibility and control – openness axes by Carl Shapiro and Hal R. Varian from their 1999 book "*Information Rules*" published by Harvard Business School Press. Although these concepts are by no means new, and have been covered by several other researchers, the authors of "*Information Rules*" nevertheless succeed in presenting their views, on the matter, with a new and fresh perspective that offers an interesting link to this thesis.

4.3.1 The performance – compatibility axis

Starting the analysis with the performance – compatibility axis, it can be identified that the introduction of a new technology can be implemented in two ways, both representing the extreme ends of the axis. A new technology can thus be introduced by either a compatibility alternative, where the new technology is compatible with older technologies or a performance alternative, where the new technology has superior performance, when compared to the older technology. (Shapiro & Varian 1999, 190-192, 206)

Both extremes have their advantages and disadvantages. The obvious advantage of the compatibility alternative is the fact that it is compatible with older technologies, thus decreasing possible switching costs for its users and allowing for an easy migration path from the older technology to the new one. The disadvantage therefore would be that by offering a technology that is compatible with older technologies, one must sacrifice performance. Hence, it can be deduced that the development of technology in this alternative is based on the evolution of technology or on incremental improvements to it. The compatibility alternative, stressing an evolutionary direction, is pictured in part A of figure 10. (Shapiro & Varian 1999, 190-192, 206)

Figure 10: Performance and compatibility axes



Source: Shapiro & Varian 1999, 191

The performance alternative, on the other hand, relies on radical or revolutionary changes to the technology, allowing for substantial increases and improvements in performance. But again, just as with the compatibility alternative, the improvement in performance does not come without a price. The clear disadvantage is having to sacrifice compatibility for the sake of improved performance. This situation typically leads to increasing switching costs for the users of the older technology seeking to move to the new technology. The performance alternative, stressing the revolutionary direction, is pictured in part B in figure 10. (Shapiro & Varian 1999, 191)

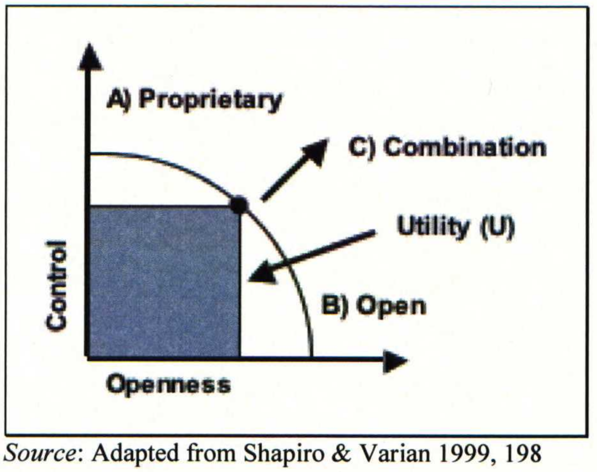
In addition to being able to choose any one of the extreme cases, one could also decide for a combination or compromise between the extremes. In this way, one would seek to maximize the benefits from both extremes, while trying to minimize the disadvantages. The new introduced technology would therefore be a mix of increased performance with reasonable backward compatibility. The operating systems upgrades offered by Microsoft, Apple and various Unix /

Linux vendors are good examples of compromises between performance and compatibility. The combination alternative is pictured in figure 10 as part C. (Shapiro & Varian 1999, 191-192)

4.3.2 The control – openness axis

The second axis, the control – openness axis, works in the same way as the previous axis. The introducer of a new technology can choose between two extremes or a combination of the two; retain complete control over the technology or adopt an openness strategy, where the technology is typically licensed to other manufacturers and vendors. Choosing the control alternative means keeping the technology proprietary and is seen to be a viable alternative if the introducer enjoys a strong market position or has power in controlling product standards and interfaces. The obvious disadvantage of the control strategy is that clients fearing lock-in will usually prefer to choose a nonproprietary technology. The proprietary alternative is shown in part A in figure 11. (Shapiro & Varian 1999; 197, 203)

Figure 11: Openness and control axes



Source: Adapted from Shapiro & Varian 1999, 198

The openness strategy becomes the best choice when no one company is by itself strong enough to dictate new technology standards and interfaces or when multiple products must work together. The introducer of open technologies often operates in mass markets, where profitability is often gained through an increase in market share. This is totally the opposite of the proprietary alternative, where markets are usually smaller, leading to closer (1-to-1) client relations. Profitability is achieved primarily through client share-of-wallet rather than market share thinking. The openness alternative is shown in part B in figure 11. In addition to being able to choose any one of the extremes, the control or openness alternatives, one could again opt for a compromise solution, namely the combination alternative, shown as part C in figure 11. (Shapiro & Varian 1999; 195, 201-205)

4.3.3 Shapiro & Varian axes modifications

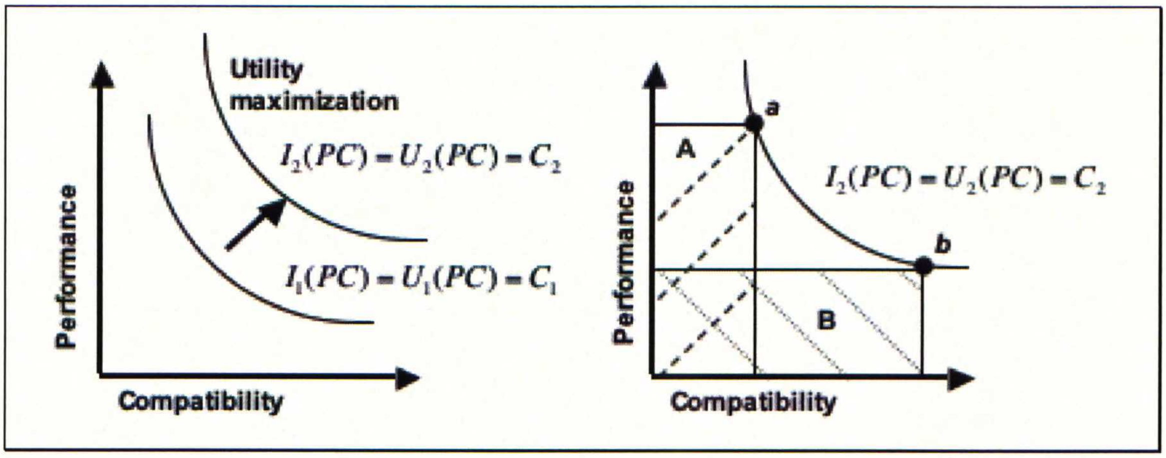
Shapiro and Varian show many interesting and insightful ideas, concepts and examples but do not present some key assumptions that should have been made. The graphical depiction of the compatibility – performance axis (Shapiro & Varian 1999; 191) shows a decreasing, concave function. The subliminal implication means that the relation of any two points on the function is not constant. Thus, it can be mathematically shown that firms should always choose the combination alternative as this would lead to the most performance and compatibility derived utility i.e. a maximization of the area between the axes and a point on the function (see part C in figure 11).

Another, alternative, interpretation of the figure would be that one could gain substantial performance improvements with only minimal sacrifices in compatibility when moving from a pure compatibility alternative towards a combination of both. Or the other way around, one could gain substantial compatibility with only a minor performance decrease when moving from a pure performance alternative to the combination alternative.

The argument that this thesis makes against this logic is very simple. It is merely stating that an assumption must be made that the relation of any two points on the function be constant. This leads to several interesting implications. Firstly, the shape of the function changes from a decreasing concave to a decreasing convex one. Instead of bowing away from the axes intersection point it now bows towards it. Secondly, change from any point on the function to any other is completely relational i.e. the substitution effect between compatibility and performance is constant. Substituting one "*unit*" of compatibility gains one "*unit*" of performance. Alternatively, a performance "*unit*" can be swapped for a compatibility "*unit*".

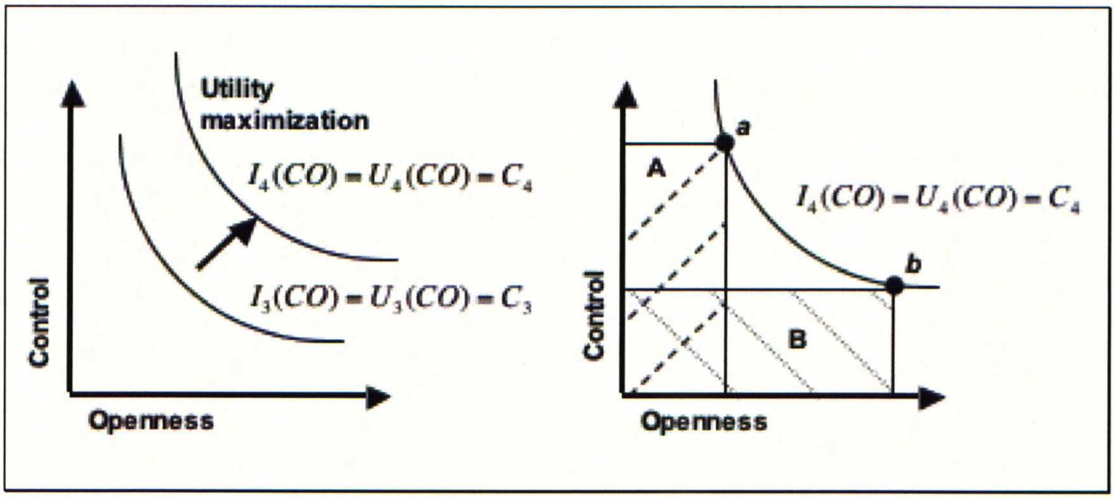
Thirdly, both compatibility and performance are equally good. Entities deciding between compatibility and performance are, by default, indifferent to the relations of compatibility and performance they choose. Fourthly, utility maximization only occurs by moving the whole function to the right, not by merely moving along the function, as Shapiro & Varian present. Finally, the shaded utility area is not constant in the old model, whereas it is on the new one. Figure 12 elaborates on these issues.

Figure 12: Altered performance – compatibility function[22]



As with the performance – compatibility axis, Shapiro and Varian do not make the key openness – proprietary function assumptions that have been presented. Again, this leads to the problems described previously. To maximize the benefits from both the proprietary and openness alternatives, companies must shift their indifference curves to the right, not merely move along the function as Shapiro and Varian claim. The following diagrams, pictured in figure 13, elaborate on these issues.

Figure 13: Altered openness – control function[23]



[22] Please note that area A is equal to area B in the right-hand part of the figure, although it may not seem so from the graphical depiction. This is due to restrictions in the drawing application used.
[23] Please note that area A is equal to area B in the right-hand part of the figure, although it may not seem so from the graphical depiction. This is due to restrictions in the drawing application used.

The key differences between the old Shapiro & Varian model and the new model are presented in table 13.

Table 13: Key differences between Shapiro & Varian (1999) and modified model

Variable	Shapiro & Varian (1999)	Modified Shapiro & Varian
Function curvature	Decreasing concave, $f''(x) \leq 0$	Decreasing convex, $f''(x) \geq 0$
Relation of any two points on function	Not constant	Constant
Maximize utility	Move along the curve (to the middle)	Shift whole curve (to the right)
Utility area	$A \neq B$	$A = B$

Source: Adapted from Shapiro & Varian 1999, 191-209

4.3.3.1 An Edgeworth box solution

While introducing some of the seminal ideas and concepts presented by Shapiro & Varian, this thesis has been concentrating on two axes and four variables. To continue this analysis, one introduces the concept of using Edgeworth boxes, borrowed from the field of micro-economics and more specifically the study of general equilibrium (Jehle & Reny 2001, 181; Perloff 2000, 315), as a possible solution to the above-presented problems and issues. This presentation finalizes and finishes the leveraging of the technology strategy component.

The initial problem with the four variables, control, performance, openness and proprietaryness, is to find an optimal position or solution to maximize their benefits, i.e. utility. It has already been shown that each of the two functions, that are made up of the four variables, act very much like indifference curves, leading to suggest the possible applicability of an Edgeworth box solution.

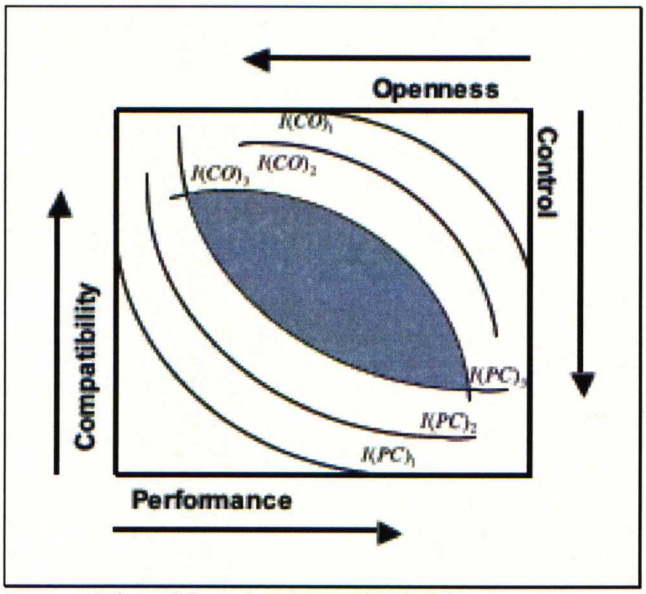
Originally an Edgeworth box is defined as a diagrammatic representation of an economy with two consumers and two goods, without any production. The theory of Edgeworth boxes states that the two consumers will trade their goods until a pareto-optimal situation has been reached, i.e. a situation where no one can be made better off without hurting the other. The final endowment point lies at a point, where both of the consumers' indifference curves are tangent to one another[24]. (Landsburg 1999, 281; Jehle & Reny 2001, 181; Perloff 2000, 316)

[24] For more in-depth explanations on Edgeworth boxes and their properties please refer to either Landsburg 1999, chapter 8 or Perloff 2000, chapter 10. Jehle & Reny 2001 provide a more advanced elaboration with mathematical notations in chapter 5 of their book.

With minor alterations and adjustments, an Edgeworth box can be put to use in determining the optimal set of positions, which maximize the utility derived from the variables. For this to happen, one would need to distinguish a balanced state of equilibrium among the variables, which is pareto-optimal. Micro-economic theory clearly articulates that an "*invisible hand*[25]" will guide the exchange between variables to this point, regardless of absolute amounts or relative positions of the variables. To proceed, the two new axes are plotted as one rectangular box.

The Edgeworth box in figure 14 now pictures all four variables on four axes. The corresponding indifference curves have also been drawn. The shaded area, where the two indifference curves first overlap, is called a region of mutual advantage (Landsburg 1999, 281), which defines a set of points that are pareto-preferred to initial states e.g. on $I(PC)_2$ or $I(CO)_2$.

Figure 14: Edgeworth box solution, stage 1



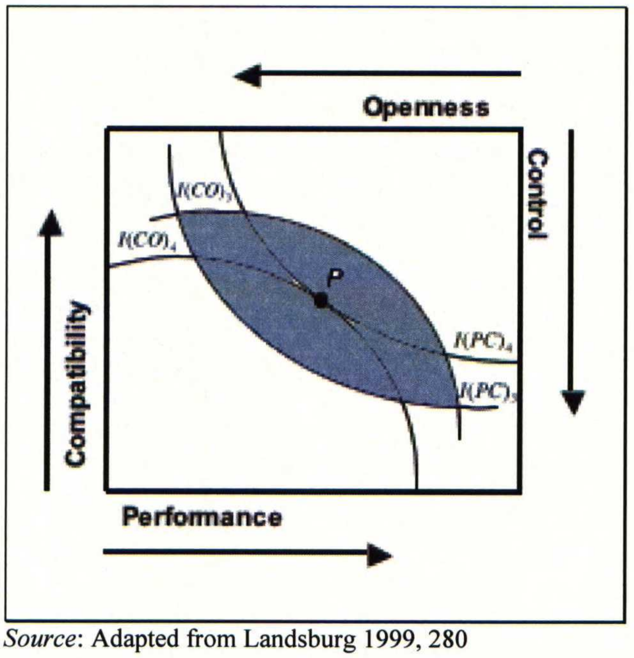
Source: Adapted from Landsburg 1999, 280

It is now known that an equilibrium solution to balancing the four variables must occur in the region of mutual advantage. Continuing on the same logic and structure, keep drawing indifference curves until two such curves exist that only overlap each other at one point, where they are tangent to one another. This happens at point *P*, which is shown in figure 15.

---

[25] First introduced by the great eighteenth century economist Adam Smith, the "*invisible hand*" is a concept used to describe trade phenomena among utility maximizing entities that inevitably end up as balanced equilibrium solutions (e.g. Landsburg 1999; 44, 279).

Figure 15: Edgeworth box solution, stage 2



Source: Adapted from Landsburg 1999, 280

Simplifying the figure of unused indifference curves, one finds two indifference curves $I(PC)_4$ and $I(CO)_4$ that are tangent at point P, which is the pareto-optimal equilibrium among the four variables.

This analysis has focused on a graphical solution to the problem, and thus does not present an exact mathematical answer. A second concern worth mentioning is that this equilibrium implies that the utility among the variables is constant, which is hardly the case in real life situations. However, these issues do not undermine the importance of using a simple graphical process of reasoning to produce a solution, which in turn can act as an important managerial decision-making aid.

4.4 Understanding the leveraged model

Earlier, in the previous sub-chapters, each individual, leveraged component has been studied and examined. The focus has mainly been on presenting the components, not on explaining why the new components work in the way they do. In this sub-chapter, the aim is to answer questions relating to the "*why*" and "*how*", to increase understanding of the new components and to present scientifically valid argumentation to support the presented claims.

This sub-chapter begins by presenting the key new drivers behind the new components, the subjects of change, and how they form and become the objects of solutions. Finally, some *preliminary*, *tentative* mathematical notations to facilitate the transformation of these components from implicit concepts to explicit formulas will be presented.

4.4.1 Subjects of change, objects of solutions

What is the fundamental reasoning behind the logic to decide that the new components work in the way they do? Why were the selected paths followed in deciding the outcome of the new components? These are just some of the questions needing answers, completing and finalizing the analysis on the new, leveraged components.

Beginning with modularity, as the key, new element in product architecture, one can see that modularity offers a way to minimize, or even avoid, complexity, and to preserve and enhance flexibility within a complex software system. Additionally, modularity has been used to consolidate design and architectural styles into one simple, easy to understand concept. Modularity facilitates flexibility and is essentially a prerequisite to the future extensions component. Modularity also allows increasing the level of abstraction to form a set of high-level design rules, more suited for management than a technical, technological perspective.

Flexibility is the seminal concept behind the new, leveraged future extensions component. Flexibility was introduced because the original Sääksjärvi model institutes extensions and extendibility in a very rigid, one dimensional way, not allowing for a great deal of movement or elasticity. Materializing as technology options, flexibility permits a range of new alternatives and options for management to choose from during the different stages and phases of platform and product design and development. Essentially, flexibility has provided a new dimension to decision-making; improving quality while remaining simple. With the decision-tree application of viewing technology options, a simple, uncomplicated process has been provided to capture the flowing logic and linear judgment needed to implement flexibility into decision-making.

With the introduction of product characteristics as the key, new driver of technology strategy, one is able to witness a clear, conscious and distinct transition from a component facilitating ex ante product platform issues to an ex post component, empowering product positioning as a result of product platforms. This transformation fundamentally changes the logic and philosophy of the component. Managers and executives need no longer focus on concerns relating to the development and design of a platform but can direct their attention to positioning their products (or product families) after product launch. Instead of concentrating on short-term tactical issues one can give attention to long-term strategic product positioning issues.

Alternatively, the view changes from an internal, micro, company specific view to an external, macro, market-specific view. To sum up, the Sääksjärvi technology strategy component is ex ante, facilitating product platform based development whereas the new component is ex post, empowering managers and executives to position their products with key characteristics. Table 14 summarizes the key points.

Table 14: Summary of change agents and new solutions[26]

Sääksjärvi (1998) model components	View of Sääksjärvi components	Agent of change	New solution
Product architecture	Guidelines for design and architectural style	Concept consolidation and complexity mitigation with modularity	Design rules as functions in a three dimensional spatial model
Future extensions	Directions for extension, vectors of augmentation	Flexibility through real option analysis	Decision tree to overall vision
Technology strategy	Infrastructure for design and development	Post launch positioning through feature characteristics	Edgeworth box equilibrium

Finally, to complete the examination, a graphical illustration of the new, leveraged model will be presented. The *primary component*, steering and guiding the use of the model is product architecture with its key, new concept, modularity. This component forms the basis for all decision-making, as the spatial model, presented earlier, and the value calculations for the six modular operators essentially allow for the capturing of the design rules and principles into a function. Modularity provides a higher level of abstraction than presented by Sääksjärvi, which should appeal to managers and executives more. This abstraction level is a high-level, conceptual guideline that managers can use to decide on design issues.

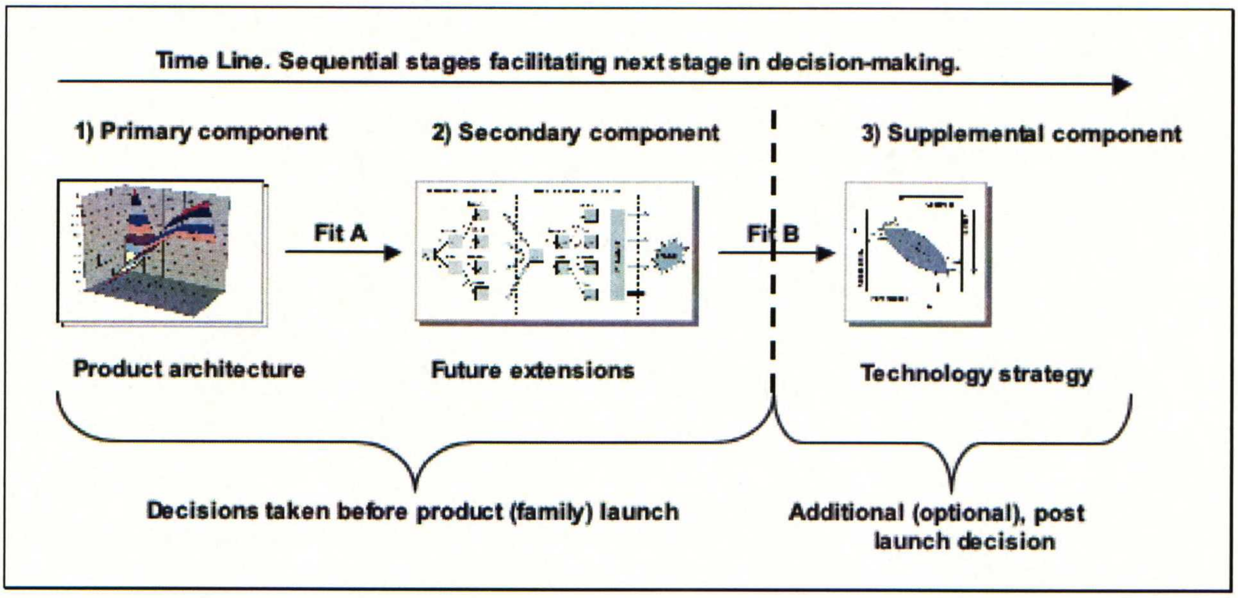
In addition to providing a means to consolidate the ideas put forth by Sääksjärvi, the element of modularity, in product architecture, provides an additional element, namely flexibility, which in turn is the foundation for the future extensions component. Categorized as the *secondary component*, the new future extensions component provides a structured, process-like flow for phased development and its investment and decision support. The decision tree approach used will allow management with added flexibility in times of technological change and market fluctuations.

[26] From a methodological point-of-view comparing the views of the Sääksjärvi components with the agents of change is not the best possible course of action, as their underlying assumptions differ. This has nevertheless been done to facilitate the move to the new solutions.

This flexibility, materialized as technology options, will be more familiar, as a top-down sequence, to management than the rigid one-dimensional, bottom-up, technology oriented approach used by Sääksjärvi. It is important to note that the fit (depicted as fit A in figure 16) between and among product architecture and future extensions is classified as very fundamental and strong. These two components are very closely related and ultimately function best as a pair.

Both product architecture and future extensions differ from the third, *supplemental, component*, technology strategy, as they are used prior to and during platform and product development, whereas technology strategy is clearly a post-production tool, used to position a product through its key feature characteristics. Contrary to fit A, one must categorize the fit between future extensions and technology strategy (depicted as fit B in figure 16) as weak, because of the changing time dimension at their intersection. The new, leveraged model is depicted in figure 16.

Figure 16: The new, leveraged model



4.4.2 Transformation from implicit concepts to explicit formulas

The analysis and examination of the new, leveraged components will continue with some mathematical notations, to provide further depth, meaning and understanding to these new components. The mathematical formulas also translate the components from a graphical form of depiction into a generic universal language. Alternatively, with these notations one facilitates change and development from implicit concepts to explicit formulas. However, it is underlined and emphasized that these notations are *preliminary* and *tentative*, as proof of the notions is not presented and because no real numerical data is presented as a basis for calculus.

4.4.2.1 Product architecture

Earlier, the three dimensional spatial model has been presented. It was formed from the six modular operators to present a visual aid, capturing the essence of the new, leveraged product architecture component. In Sääksjärvi's view, the architectural style and design are the core of product architecture, whereas this thesis has put forth the notion of modularity as the seminal concept. Note that these two differing views are not contradictory or mutually exclusive, but rather complements enhancing each other. Next, some mathematical notations are assigned to each of the six modular operators, as a basis for value and future payoff calculations.

Assuming that design is an evolutionary, value-seeking process and that the six modular operators can be applied at many points and in different combinations, over time this value-seeking process leads to ever more complex, diverse and dispersed modular systems (Baldwin & Clark 2000, 246). With these calculations, this thesis seeks to formulate a basis for conducting value assessments.

Module performance is defined by partitioning overall system performance into system-level value, S_0, and n module value, $X_1,...,X_n$. Because of S_0, the value of the system is always greater than the sum of its parts. Furthermore, every random variable X_n, corresponding to a module value, has a normal distribution. This leads to an additional constraint, as the expected value, $E(X_n)$, is always zero. Finally, for a module design with n tasks, the variance of its value, σ_n^2, increases linearly with the number of tasks. Therefore, $\sigma_n^2 = \sigma^2 n$ must hold. (Baldwin & Clarke 2000, 254-255)

Given these assumptions, the value of a one-module design can be calculated. Denoting:

$$V_1 = S_0 + E(X_n^+) \tag{3}$$

(Baldwin & Clarke 2000, 255)

where V_1 is the value of a one-module design and the superscript '+' means that the expectation only applies to outcomes above zero. The equation above, therefore, essentially means that the final outcome of the design process is a random variable with a normal distribution, a mean of zero and a variance of $\sigma^2 n$, that the designer implements only if its value is superior to the old one[27]. (Baldwin & Clarke 2000, 254-257)

Splitting and substitution

The value of splitting a system into j modules is simply an extension of equation 3, written as:

$$V_{split,j} = S_0 + E(X_1^+) + E(X_2^+) + ... + E(X_j^+) \tag{4}$$

(Baldwin & Clarke 2000, 259)

If a new module design has a value greater than zero, i.e. superior to an existing one, the module will be incorporated into the system. As for substitution, where the designer can swap one module for a better version of the same module, one denotes n tasks with k parallel, independent design efforts. For a random payoff function $\tilde{X}$ and the value of the best of k designs, a function $Q(\tilde{X};k)$ will be used. The calculation for the value of substitution becomes:

$$V_{subs}(\tilde{X}_1 ... \tilde{X}_j; k_1 ... k_j) = S_0 + Q(\tilde{X}_1; k_1) + ... + Q(\tilde{X}_j; k_j) \tag{5}$$

(Baldwin & Clarke 2000, 264)

(Baldwin & Clarke 2000, 247-264)

Augmentation and exclusion

In a modular, complex system, designers are not limited to splitting modules and improving these through substitution. They have additional options with augmenting and excluding modules. Every module has its own value function, which stores information between complexity, visibility and technical potential. This value function has a peak or maximum reflecting the optimal amount of experiments worth trying on the module in question. Augmenting a modular, complex system adds a new module i.e. a new source of value to the system. Writing:

$$V_{aug} = \max \delta(n_v)^{1/2} Q(k_v) - cnk_v - Z_v \tag{6}$$

(Baldwin & Clarke 2000, 304)

where k represents the optimal number of trials for the module in question, depending on the module's technical potential, δ, its complexity, n, the cost of experimentation, c, and the cost of visibility, Z. Equation 6 essentially indicates that a new module has the same ability to develop

[27] The formal mathematical notation would be $E(X^+) = \int_0^\infty Xf(X)dX$, where expected performance is calculated by weighing each potential outcome by its probability and summing up. (Baldwin & Clarke 2000, 265)

and evolve through repeated, continuous experimentation and substitution, as do existing modules. (Baldwin & Clarke 2000, 302-306)

Exclusion is the obvious and logical operator opposite augmentation, working to reduce cost by either letting end-users or designers omit modules from their individual configurations and systems. Exclusion is more complex as a concept because it functions on two levels, system and modular. One writes:

$$V_{excl} = \max(S_o + \Sigma V_{subs} + \Sigma V_{aug} - c) \tag{7}$$

(Baldwin & Clarke 2000, 311)

where c is design cost. (Baldwin & Clarke 2000, 307-314)

Inversion and porting

Inversion requires the existence of modules within modules. Therefore, inversion can only take place if a system has been split at least twice. Baldwin & Clarke (2000, 327) argue that the best time to apply the inversion operator is when the costs of redundancy first begin to outweigh the benefits of experimentation. Writing:

$$V_{inv}(j,k) = cnl - C_{inv}(j,k) - \delta_{inv} n^{1/2} Q(m) - Z_{inv}(m) \tag{8}$$

(Adapted from Baldwin & Clarke 2000, 327)

Where cnl is the cost savings in hidden modules' experiments, $C_{inv}(j,k)$ is the cost of designing module j with k number of experiments, $\delta_{inv} n^{1/2} Q(m)$ is the value lost in hidden modules' experiments and $Z_{inv}(m)$ is the costs of visibility. (Baldwin & Clarke 2000, 323-329)

Porting is simply the action of taking a hidden module from a system and making it work in another system. Porting requires three steps. An initial split of the module into system-dependent and system-independent functions. Next a representation for the system-dependent parts is required, specifying e.g. design rules and hidden modules. Finally, a set of translator modules must be designed to make the portable system compatible with other systems. Denoting:

$$V_{port} = V(M;S) + (M-1)C_{os} - MC_{trans} - C_{boot} \tag{9}$$

(Baldwin & Clarke 2000, 344)

where *V(M;S)* is the value of being able to switch among M systems at a cost of S, $(M-1)C_{os}$ is the cost savings with not having to redesign each operating system from the start, MC_{trans} is the cost of designing the required translator modules and C_{boot} is the cost of "*bootstrapping*" i.e. the cost of developing a representation of the system-independent portable module. (Baldwin & Clarke 2000, 338-344)

4.4.2.2 Future extensions

Previously, an evolutionary, two-stage depiction of future extensions as the new, leveraged component has been presented. The force driving this evolution is the initial product vision that also provides a goal for all extensions. Therefore, extensions are not carried out on a tactical level, addressing new market segments or product versions, but rather on a strategic level, mapped by a product vision. To operationalize and validate these initial thoughts, one refines the new future extensions concept by assigning simple numerical examples. This presentation is based on pages 121-124 of Copeland & Antikarov 2001.

The methodology used depends on the characteristics of the underlying asset and on the features of the option that is contingent on it. For simplicity, it is assumed that the value of the underlying asset follows a multiplicative[28] series through time and that the underlying asset does not pay out cash flows (dividends). Denoting as follows:

Value, $V(P_0) = X$, at the beginning of lattice platform evolution

Improvement, $I > 1$

Finalization, $F < 1$, calculated as improvement^{-1}

$p = 0.5$ and $q = 1\text{-}p = 0.5$

If improvement = 1.1, finalization = 0.90909 (=1.1^{-1}) and $V(P_0) = 100$, then by recombining this binominal decision tree, four points of interest are found (Copeland & Antikarov 2001, 122):

1) Every even-numbered time period (0, 2, 4 etc.) has a middle point value of exactly 100
2) Every odd-numbered time period (1, 3, 5 etc.) has a geometric average payoff of 100
3) In the uppermost branch, the value approaches positive infinity (although with zero probability), as the number of time periods approaches infinity. Lim $I^T = \infty$, as $T \to \infty$.
4) In the bottom-most branch, the value approaches zero (although with zero probability), as the number of time periods approaches infinity. Lim $F^T = 0$, as $T \to \infty$.

Therefore, as the number of time periods becomes very large, the distribution of outcomes at the end branches approaches a lognormal distribution function. The table 15 elaborates these issues.

[28] The value of a multiplicative time series is always equal to or greater than zero, as opposed to the value of an arithmetic time series, which can be negative.

Table 15: Multiplicative stochastic process for platform evolution stage

Time period 0, T_0 → →	**Time period 1, T_1 → →**	**Time period 2, T_2**
		$p^2I^2 * V(P_{2.2}) = 121$
$V(P_0) = 100$	$pI * V(P_{1.1}) = 110$	$pqIF * V(P_{2.1}) = 100$
	$qF * V(P_1) = 90.91$	$qpFI * V(P_2) = 100$
		$q^2F^2 * V(P_1) = 82.64$

Source: Adapted from Copeland & Antikarov 2001, 122

These calculations can be extended with subsequent time periods or with the product (family) evolution stage. The calculations above are static in nature but could just as well be converted into dynamic, with different values or constants for each time period. With this change, the distribution would no longer be lognormal.

4.4.2.3 Technology strategy

From earlier presentations, Edgeworth boxes and their pareto-optimal equilibrium solutions should be familiar. Although the Edgeworth box as such can be very useful, as a decision-aid tool, this thesis nevertheless seeks a deeper understanding with the following mathematical notations. First one denotes four variables, their respective functions and constraints:

v_1 = variable 1, performance, where $v_1 \geq 0$

v_2 = variable 2, compatibility, where $v_2 \geq 0$

v_3 = variable 3, openness, where $v_3 \geq 0$

v_4 = variable 4, control, where $v_4 \geq 0$

$f(PC) = f(v_1, v_2)$

$f(CO) = f(v_3, v_4)$

C = constant

W = weight, where $\sum_{i=1}^{4} w_i = 1$, and $w \geq 0$

The relation between the constant C and the weight W is written as

$$\sum_{i=1}^{4} w_i v_i = C \qquad (10)$$

to satisfy the constraint of constant total utility[29]. Finally, the initial problem of maximizing derived utility from the four variables simplifies to solving P. As previously discussed, the Edgeworth box solution solving P is constant. Therefore, writing

$$\max(U)\sum_{i=1}^{4} w_i v_i = C \tag{11}$$

shows that a solution at P is always constant C. P is then simply solved by finding equal derivatives from both indifference curve functions. Table 16 shows three alternative ways of solving P.

Table 16: Three alternative notations for solving equilibrium solution *P*

Equilibrium Solution	Description	Equation
P	Derivatives from both CO and PC indifferences curves are equal	$I'(CO)_k = I'(PC)_k \quad (12)$
	Alternative notation based on functions of CO and PC	$\frac{df}{d(CO)}(CO) = \frac{df}{d(PC)}(PC) \quad (13)$
	Alternative notation based on variable pairs	$\frac{df}{d(v_3,v_4)}(v_3,v_4) = \frac{df}{d(v_1,v_2)}(v_1,v_2) \quad (14)$

[29] Essentially, this means that tradeoffs between the four variables are played as a zero sum game i.e. increasing one unit decreases another. Total utility (or payoff) is always constant.

5. CASE: EVOLVIS EURO RSCG

In the previous chapters, the Sääksjärvi product skeleton model and its transformation and adaptation from a tactical, technology driven, software product development model to a business driven strategic decision support system has been examined and discussed. To apply and test the new leveraged model in a real life business context, empirical research must be conducted on a suitable case company to evaluate the theoretical findings on. This chapter first presents the case company and its software product development. Finally the new leveraged model is applied to the case company.

5.1 The case company

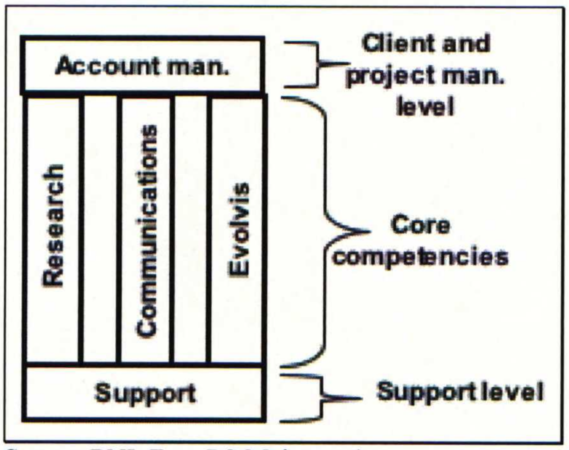
5.1.1 BNL Euro RSCG Oy

BNL Euro RSCG Oy (BNL) is one of the leading Finnish communications and PR consultancies. Founded in 1986, BNL is 70 % owned by Euro RSCG Worldwide, a network of advertising, marketing, communications and interactive agencies listed on the New York and Paris stock exchanges. In addition to the four managing partners, who own the remaining 30 %, BNL employs around 45 professionals. In 2001 BNL's turnover was FIM 36 million with a 15 % operating profit. (www.bnl.fi, 16.09.2002)

BNL provides its clients services ranging from communications research, strategic planning, crisis communications, public relations, public affairs, corporate communications, and investor relations to print publications. Over the years, BNL has had profitable client relations with companies from such diverse industries as forest and paper, transport, healthcare, oil and energy, telecommunications and software, retail, food and beverage, banking and insurance and defense, as well as with various governmental and municipal organizations. (www.bnl.fi, 16.09.2002)

During spring 2001 BNL's management identified a distinct gap in the company's service portfolio. Thus far, BNL had lacked expertise in the field of digital communications, which had been brought to attention, on several occasions, by employees and clients alike. Being one of the market leaders in its respective business sector, BNL had to be a full service consultancy, without the need to rely on external partners in key service areas such as digital communications. Increasing pressure to move and extend traditional communications to incorporate a digital presence and form, led to the establishing of BNL's own digital communications unit, Evolvis Euro RSCG (Evolvis), in May 2001. BNL's organization structure is presented in figure 17.

Figure 17: BNL Euro RSCG organization structure



Source: BNL Euro RSCG internal company presentation slides, 23.09.2002

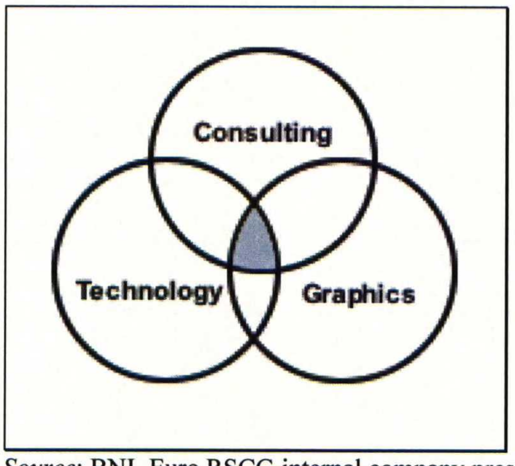
5.1.2 Evolvis Euro RSCG

Today Evolvis is an independent business unit of BNL[30]. The business strategy of Evolvis is to support and extend the traditional communications and PR activities designed and produced by BNL for its clients. Evolvis employs 12 professionals, divided into three groups of four employees each: consulting and project management, graphics, and technology. The expected turnover for Evolvis during 2002 is approx. € 0.6 million with an operating profit of about 10 %. (www.evolvis.net, 17.09.2002)

Evolvis executes its business strategy by using the most effective and appropriate digital technologies and media for each case and project. The main medium in use is the Internet (and its variations, Intranets and Extranets), with a steadily growing importance in mobile communications and digital television. The bulk of Evolvis's everyday business is made up of designing and producing Internet sites and web pages that support the desired communications plans and activities. For example, if BNL handles the corporate communications account for a client, Evolvis's job would be to plan and produce the corporate website. (www.evolvis.net, 17.09.2002)

[30] Evolvis is not an independent company from a legal point-of-view. The name Evolvis Euro RSCG is merely a marketing name used to distinguish Evolvis from BNL and emphasize the slight differences in their respective core competences.

Figure 18: Evolvis Euro RSCG internal organization structure



Source: BNL Euro RSCG internal company presentation slides, 23.09.2002

Figure 18 portrays the three competence areas of Evolvis, which also equal the internal organization structure. The three components overlap each other forming a core, central competence, depicted in the darkened triangle in the center of the figure.

Specific Evolvis related services include digital communications consulting and strategy planning, crisis and issues management and monitoring, user interface design, website production and updating and maintenance services. It is to be noted that Evolvis does not sell or produce software products or services by themselves, but uses and applies them as tools for the above-mentioned services. Software tools in use include e.g. databases, middleware solutions, monitoring applications and agents, content management solutions and authoring tools.

5.2 Current product development

As noted previously, Evolvis is not a software company. It does not produce software directly for its clients, but rather develops and uses software applications as tools to promote communications and PR activities. This has many implications for product development, both positive and negative. Thus, from Evolvis's point-of-view, software is a means to an end, not an end in itself.

Starting with the positive aspects, one should note that not selling software directly to end-users can streamline the development process in key areas such as documentation and product marketing, essentially allowing Evolvis to cut a few corners here and there. From a negative perspective, having to organize for product development in a consultancy business model can be challenging for management and employees alike.

Another key feature of Evolvis product development and technological direction is the commitment to open source software. Naturally, this changes the business dynamics, when compared to business models based on a proprietary software and technology strategy. In the following sub-chapters these and other related issues will be discussed.

5.2.1 Organization

The organization of product development centers around the technology team of Evolvis. Currently the team consists of four talented, software engineers, with several years of systems design and programming experience. Evolvis management, the group manager and the senior consultant (the researcher) make up the remainder of the product development organization.

The product development organization lacks formal characteristics such as explicit roles and responsibilities and chaired meetings etc. Communications and decision-making is handled informally during everyday activities. Management control over the unit is virtually non-existent for everyday operational issues. Only larger, complex and far-reaching decisions are dealt with collectively, with a case-by-case method. Therefore, the product development organization is very much inline with the overall organizational philosophy of Evolvis, which is to be a relatively small, nimble and efficient team that can quickly adjust to changes in technological environments and client expectations. Thus, the organization of product development is not based on a "*laissez faire*" attitude, albeit it may seem so to the casual outsider, but rather on a specific, well articulated decision.

Being a small, four-man unit, the product development team does not have a structured hierarchy, but rather an evolved and sophisticated order of seniority based on capabilities and experience in a given area. The unit functions very much like a miniature open-source software development community. Receiving a market-price compensation package is not the main reward system, but rather the acknowledgement and approval from ones peers.

5.2.2 Methods and tools

The whole technology philosophy of Evolvis is directly derived from its strategy and organization. When Evolvis was founded, the key question regarding technology was not e.g. which technologies to use and how to implement them in the most efficient manner, but rather firstly to acquire some of the best programmers and systems engineers in the field and secondly to ensure their organizational fit. Following this, a collective decision on the most appropriate technologies to use and implement would be made. (Evolvis business plan, version 1.2, 21.03.2001)

Today, the driving technological force in Evolvis is the commitment towards the principles and practices or open source software development. The potential benefits of open source software development can be substantial when compared with proprietary, closed, software development. Open source allows the leveraging of and learning from existing (open source) technologies and applications. In addition, it supports the business strategy of Evolvis, which is not to be a software product company, but rather to provide value added services based on software. The obvious disadvantage of the use of open source software is that Evolvis is unable to limit the use of internally developed applications from its competitors. So far, this has not been a problem, as the adoption of open software always carries an implicit cost in the form of an adoption function.

Using open source software naturally leads Evolvis to use the most common methods and tools adopted by the larger development community. Therefore, the architecture in place is a LAMP[31] environment. As for specific development methods or tools, none are currently in use, although some experiments with e.g. UML are underway. "*Extreme Programming* (XP)" (see e.g. Beck 2000) is a development philosophy that best describes the way in which programming and development work is conducted, although the Evolvis way is by no means a pure XP implementation.

5.2.3 Software product development process

Despite the fact that Evolvis functions in a very organic and non-structured way, some of the key elements in the processes of decision-making and product development have been identified. As with many Evolvis specific organizational features, these processes have no formal basis and are not executed by a predefined set of principles. The processes are based on everyday working procedures that one would find in any organization. Although decision-making in Evolvis can be described as fast, the actual process is nonetheless based on iteration cycles and repeatedly turning to previous parts of the process for insight and reconsiderations.

Both the decision-making and development processes have distinct characteristics that distinguish them from one another and from other processes. Firstly, the decision-making process is characterized as a support process for other processes, because by itself decision-making has no

[31] Linux (operating system), Apache (webserver), MySQL (database management system) and Perl / PHP (programming and scripting language).

value. The facilitation and provision of support for other processes is the only function in generating value for the decision-making process. Therefore, it is very difficult to model the decision-making process, as it is an integrated part of all other processes. The development process, on the other hand, is characterized as a core process, because it holds value to the overall business of Evolvis and is a process without which the company could not operate.

5.2.4 Application portfolio

Currently, one of the main focuses of Evolvis is consolidating its key software modules and individual applications into a formal, consistent structure, namely a portfolio or suite of applications. Until recently, all software modules and applications have existed as separate entities, without much order and alignment, arising from a myriad of backgrounds and histories, of either individual development projects or as part of client projects.

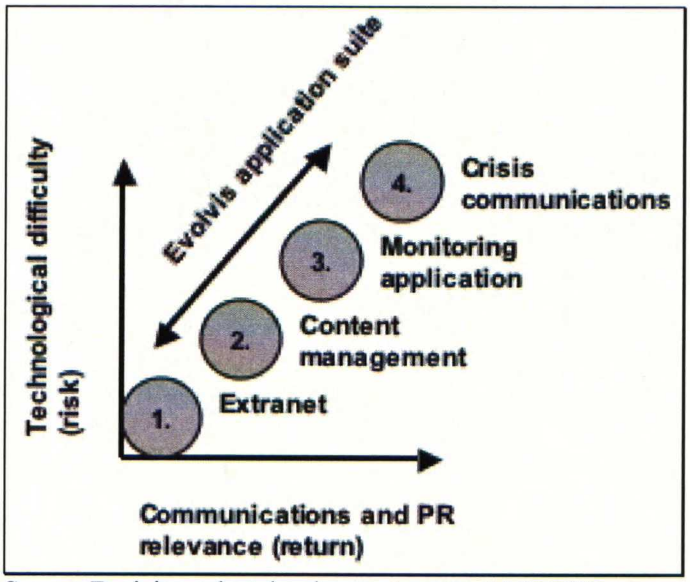
With the development of an application portfolio, Evolvis hopes to achieve the following advantages for its business. First, all applications will be built around a common core, the platform, reducing development and maintenance costs and time, while simultaneously leveraging existing designs and code. Second, the application portfolio will allow Evolvis to differentiate itself from its competitors, who tend to position themselves either as pure consultancies or as software developers. The application suite will allow Evolvis to achieve some of the advantages from both business models. Third, the underlying pricing mechanisms that dictate the formation of fees invoiced from clients will have more freedom of movement, by allowing, on the one hand, standard hourly or daily based consultation fees and, on the other hand, fixed software license and maintenance fees.

From a technical perspective the application suite will be built around a common application server platform, Midgard[32], which is freely available with source code. Without going too deep into the reasoning behind the choice and decision to adopt Midgard as the main development platform, it is briefly stated that e.g. database connectivity, authentication and authorization, an integrated development environment, the possibility for include other open source or commercial add-on modules and an active open source development community were among the key factors influencing the decision. (Evolvis product development memo, 19.11.2001)

[32] For more information on Midgard and its technical features and abilities, please see www.midgard-project.org.

At its current development stage, the Evolvis Application Suite (EAS) consists of four key software applications categorized by their level of technological complexity and communications project relevance. The four applications, extranet, content management, monitoring application and crisis communications differ substantially from one another by function (what they do) and by operation (how they work), which places additional constraints and difficulties on their development and management. The applications are shown in figure 19.

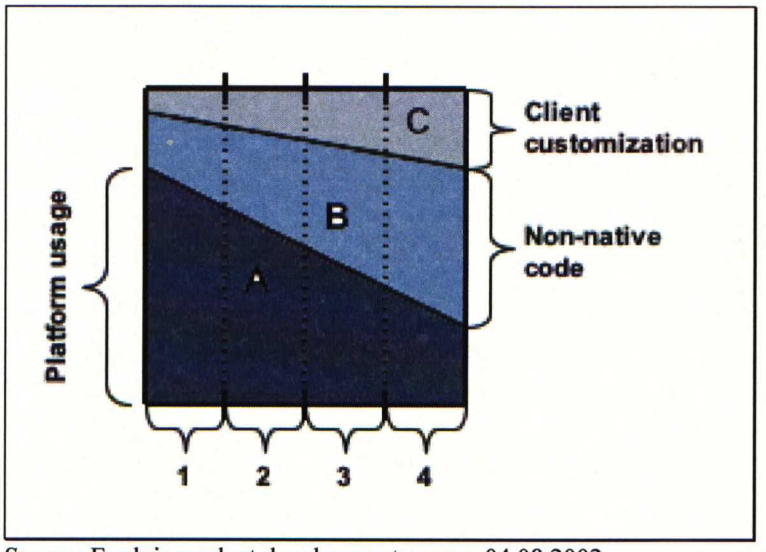
Figure 19: Evolvis application suite, EAS



Source: Evolvis product development memo, 04.08.2002

Another method of EAS application characterization would be based on the level of platform use as a percentage of the total application. Area A in figure 20 depicts this level. Area B portrays the amount of non-platform native tools and code that are needed for the application. Area C shows the amount of client specific customization that must be incurred with each implementation. The numbers, 1-4, correspond with the ones in figure 19.

Figure 20: EAS level of platform use



Source: Evolvis product development memo, 04.08.2002

It is clear from figure 20, although it is only a casual graphical interpretation of the situation, that of all the applications, the extranet is the one that uses the most out of the initial platform, and only very little non-native code and client customization. The other applications show decreasing levels of platform usage, with simultaneous increases in non-native code and client customization levels.

5.3 Applying the leveraged model

Prior to the application of the new, leveraged model, a key premise affecting the empirical study must be specified, namely that Evolvis already has an implemented product platform in use, which *did not* result from this study. Therefore, one is unable to study the effects of whether or not a platform should be used, but rather is restricted to the study of its implementation, application and its possible future extendibility. Please note that the application of the model was not based on a formal quantitative survey. It has been formulated based on qualitative information and findings from both the formal and informal interviews. The implications from this choice of method are discussed in the final chapter of this thesis.

5.3.1 Individual component applications

The application of the model begins by studying its individual components. This helps to facilitate a gradual understanding of the model and the way it functions and operates. Next the application expands to that of the whole model.

5.3.1.1 Product architecture

Product architecture, the core of the new, leveraged model, was the first component examined in the case environment. A good, strong fit was found in relation to the future extensions component, whereas a much weaker one was found with technology strategy. The graphical depiction of the component did not function as well as the mathematical notations, which seemed to work in a very coherent and consistent manner. Without these notations the component would have had only a very poor level of internal stability.

The three-dimensional model used to portray the six modular operators was not used in the decision-making process of the case unit because this would have required an exact function, which was impossible to produce. However, the simple mathematical formulas were of great help by providing a practical set of tools to use in decision-making modeling. When assessing the information type needed, the graphical depiction would have required exact, explicit information to produce a function or plane in the three dimensional space. The six modular operators, on the other hand, only require a few variables for meaningful use and interpretations, which can be easily acquired.

The conceptual, graphical depiction of the product architecture component could have been used on both a generic and specific level to support decision-making but this, again, would have required exact knowledge to form the relevant function. Therefore, this component does not align well with the overall decision-making process in Evolvis. The mathematical calculations can be used at the beginning stages of product development to assess the values and expected payoffs from different design principles.

With the system interface metric, the component fails, on a graphical level, to fully integrate with the decision-making process but succeeds on a mathematical level. The reasons for this have been discussed above. Clearly the graphical depiction has great potential, with the correct function, to not only consolidate the design principles in product development but also to compare different strategies and tactics. Based on the above observation, it is clear to state that the graphical form of the new component is not able to realize the expectations and potential originally associated with it. Despite this shortcoming, the mathematical notations are a positive surprise as they produce meaning and results above original expectations.

Table 17: Modularity and the six metrics in product architecture

Metrics / Variable	Internal stability	External applicability	Information type	Decision processing	System interface	Communication interface
Modularity on conceptual level	Strong fit with future extensions component.	NA as an exact function would have been needed.	Requires exact, explicit information to produce 3D function.	Poor alignment due to missing information for correct function.	Failure to integrate with decision-making process.	Great potential but currently lacks arguments for real use.
Explicit calculus on modularity	Holds the component together → fair stability.	Good applicability as a set of practical tools is provided.	Each operator requires a few variables of information.	Use at beginning stages of product development.	Good level of success in practical applicability.	Set of mathematical equations proves better than expected.

5.3.1.2 Future extensions

The second component under examination was future extensions, which has previously been shown to consist of two main stages, platform and product (family) evolution. Both stages support the move towards the overall strategic vision of the company. Here the initial movement i.e. vertical or horizontal, as presented by Sääksjärvi, is not relevant. Rather, the decision tree solution forms a vector of direction aligned with and targeted at a state articulated by the vision.

The internal stability of the new component proved to be good, on a conceptual level, as the graphical depiction of decision trees and the flowing nature of its internal logic seemed to be familiar. The targeting of a vision provided a consistent and coherent argument steering and guiding development decisions. The element of flexibility proved complimentary in relation to modularity and product positioning. As for explicit calculus, the differing nature of discounting future states and assigning their present values exceeded the comprehension of all but a few Evolvis employees.

Options are a useful way of incorporating flexibility into investment decisions. The new component was used primarily at the beginning stages of platform and product development to provide preliminary and tentative support for investment decision-making, resulting in a good level of external applicability. Although the formal notions of supporting investment decisions do not fit in with the informal culture and organizational structure of Evolvis, the component fits in well with the organization, as it provides a new perspective and viewpoint to valuing investments. Additionally, the introduced concept of flexibility has a perfect alignment with Evolvis' strategic objectives. Calculations on technology options were only discussed with Evolvis management.

Understanding and using technology options does not require specific data or information to work. The only requirement is staging the development into smaller and more detailed phases and their articulation with version numbering. Alternatively, the multiplicative stochastic process would require exact numerical data to provide a meaning and useful quantitative value. The new component aligned well with the initial, beginning stages of Evolvis' decision-making process. The major impact the component produced was in pointing a direction for product and platform development, rather than providing constant support during each phase.

When testing and examining the communications and integration issues related to the new component, a good fit was found for both, as the flexibility concept related to the new component communicated well and also integrated with company structures and policies. The decision tree solution was found to communicate the element of flexibility through options well to Evolvis's employees. Based on this, one can conclude that the new, leveraged component works best as a graphical depiction. As with all of the other new components, the mathematical notations did not achieve a satisfactory level because of a lack of real numerical data.

Table 18: Technology options and the six metrics in future extensions

Metrics / Variable	Internal stability	External applicability	Information type	Decision processing	System interface	Communication interface
Decision tree analysis on conceptual level	Good overall fit. Easy to understand internal logic.	Good applicability into the starting phases of a decision process.	No specific data or information required, except for detailed phasing.	Alignment with and restricted to beginning phases of decision-making.	Concept of flexibility key agent in communication and integration issues.	Central message is portrayed well. Works best on graphical, conceptual level.
Explicit calculus with multiplicative stochastic process	The incorporation of a time dimension → Differing nature of calculus.	Only used by management. No need to present calculus to remaining employees.	Data required for option value calculations.	Exact numerical data required for relevant calculations.	NA without actual, real data.	NA without actual, real data.

5.3.1.3 Technology strategy

To end the examination of the new, leveraged model's components, a look at technology strategy and how it can be assessed, as an individual entity is needed, on both a conceptual and explicit level. Beginning the analysis with internal stability, a good level of fit with the other components of the model was found, which portrays a balanced and complimenting unit, acting consistently and unanimously towards the overall benefit of the complete model. This research project was able to apply the new product characteristics to the case company because they facilitated the initial transition from an ex ante technology strategy focusing on the design and implementation of product platforms to an ex post technology strategy that empowers management to competitively position their products in relation to others in the market. Therefore, the new product characteristics worked well, as the case company already had an implemented platform for its application offering.

The conceptualization of the four variables in their Edgeworth box solution seemed to work best on a managerial level i.e. with those who are familiar with intermediate micro-economics, resulting in an above average organizational fit. As for explicit calculus examples in both internal stability and external applicability, a clear division into two separate parties became apparent; those with formal academic backgrounds found the simple derivative solution intuitive and those without high school degrees, i.e. graphics designers, preferred the conceptual depiction.

The new component did not require additional data or specific information to work. Despite this apparent benefit, the cost – benefit ratio cannot be assessed, as the component does not produce an unambiguous outcome. The component aligned itself well with the overall decision process in the case company, but did not fit into any designated phase. Based on this, one can deduce that the component works best when used to supplement and compliment decisions.

Integration of the new component into Evolvis' decision-making process proved simple and easy at first but as the "*newness*" of the component weakened it was soon forgotten. The concern is that because the new component is still a work in progress the case company does not see it as a truly rigorous and tested model. The diffusion of models takes time and it would be overly optimistic to expect a new, untested model to be adopted without any difficulties.

When used to communicate product development related questions and issues to random BNL employees, the new component seemed to enjoy a decent level of success. However, no far-reaching conclusions can be made from this, as it is unlikely that random BNL employees would admit to not understanding the model. The mathematical notations were not tested.

Table 19: Product characteristics and the six metrics in technology strategy

Metrics / Variable	Internal stability	External applicability	Information type	Decision processing	System interface	Communication interface
Edgeworth box solution on conceptual level	Good fit, consistent functioning. Complimentability.	Worked well because of ex ante / ex post transition.	No additional data or specific information needed. Cost – benefit ratio NA.	Alignment on generic level. Not suited for a specific stage.	Initial excitement → soon forgotten.	Good communication and interaction. Works best on graphical level.
Explicit calculus with Edgeworth box equilibrium	Consist with other component notations. Solving derivatives simple.	Division of case company into two camps based on education level.	No numerical data provided.	Not suitable for decision support. Requires numerical data.	NA. Calculus is not generally used in decision-making.	Not tested. Mathematical software program solves equation.

5.3.2 Complete model application

Having studied and examined each individual component against the six metrics, this research moves forward by increasing the level of abstraction to that of an overall synthesis of the new components to the three constructs.

Of the three new components, the importance and central nature of product architecture has been stressed. Simultaneously, it must be stated that of all the three components product architecture is perhaps the most problematic and complicated to design and formulate. This research project has not succeeded in producing a component with a good, or even fair, level of structural firmness. The key implication from this is that the existence of the complete model is at stake as the foundation onto which it is built is not stable and firm. With regard to functional convenience, one is unable to assess the collection of information required and therefore also the processing of decisions as it is virtually impossible to formulate an exact equation for the model in the component. Without this function, it is impossible to have interfaces between the other components in the model or its end-users in an organizational context.

As for the new future extensions component, with technology options as its key element, a fair level of structural firmness was witnessed, as the component showed a solid and stable structure. It did not, however, integrate well with the company's decision-making, especially on a numerical level because of the incorporation of a time dimension. Functional convenience proved difficult to assess, as no specific data or information was needed for the component, apart from the need to phase development into smaller portions. Representational delight worked well, especially with the graphical depiction of the component.

Technology strategy, has a good level of structural firmness, as its new set of feature characteristics are not only internally stable and consistent but also externally applicable. However, technology strategy does suffer from the different time dimension associated with it. The component requires additional information to work but the cost of this information is acceptable to its benefits. Integration of the component with the decision-making process achieves a good success rate. Finally, the representational delight factor of the new technology strategy component is excellent. The above-presented information is summarized in table 20.

Table 20: Leveraged model components and the three constructs

Constructs / Leveraged model components	Structural firmness	Functional convenience	Representational delight
Product architecture with modularity	Poor. Mathematical notations provide some support.	NA. Exact mathematical equation would be needed.	Great potential but ultimately at best fair.
Future extensions with technology options	Fair. Solid and stable structure.	No specific data or information needed. Phase development into smaller portions.	Good, especially with graphical depiction.
Technology strategy with feature characteristics	Good despite differing time dimension.	Good. Acceptable cost-benefit ratio of information. Good decision-making process integration.	Excellent. Simple graphical and mathematical solution.

Based on the information above, this research concludes and summarizes the following. Future extensions and technology strategy work best as conceptual graphical depictions, whereas product architecture works best as mathematical formulas. Structural firmness in all three components is on average fair, functional convenience has been difficult to measure and representational delight is clearly the strongest construct.

6. RESULTS, RECOMMENDATIONS AND DISCUSSIONS

Over the course of this study, the Sääksjärvi (1998) model has been presented and how it acted as a solid foundation for an attempt at providing a leveraged solution, taking the shape of a new, conceptual model, to increase the quality of managerial decision-making. Based on the initial theoretical contributions, the new, leveraged model has been brought into a business environment to test and analyze its meaning and effect in the selected case company, Evolvis Euro RSCG. The results of this thesis will be discussed on two levels. First on a theoretical level, then on an empirical one. Next, some points on the feasibility, applicability, validity and reliability of this study will be presented. Finally, this chapter ends with some issues and concerns relating to this thesis.

6.1 Results from study's theoretical part

Originally the aim of this thesis was to produce a new, leveraged model to help increase the quality of managerial decision-making in software product development. As will be discussed in the next chapters, one is unable to present valid, reliable results as to the generic success and applicability of the new model. At the same time, one is not able to dispute the possibility that the model could not help in increasing decision-making quality. Therefore, one is forced to state that a new, conceptual, preliminary and highly tentative, leveraged model has been formulated, as expressed throughout this study. Its possible success and merit cannot be assessed but it is clear that the new model cannot decrease decision-making quality, which leads to suggest its possible use, continued development and future refinement.

Answering the initial theoretical research question[33], one is forced to conclude and confess the lack of a clear answer. Surely a product platform incarnation *could* be of help; only the "*why*" and "*how*" remain unanswered, partly because of excessive ambiguity in the results. For the second research question, this thesis introduced the concepts of modularity and flexibility as key new elements that could be incorporated into the original model to increase its effectiveness as a decision support tool. However, with these new concepts, the model is by no means complete. It requires additional research and further testing to produce a clear-cut answer.

[33] For a recap of the original research questions, please refer back to page 5.

The third research question concerned the transformation of the new concepts into mathematical equations. The answer to this question is straightforward and simple; a great deal of further work is needed, as only preliminary and tentative notations have been put forth. Finally, table 21 presents a summary of the new, leveraged model in comparison to Sääksjärvi's (1998) model. Simultaneously it answers research question 4.

Table 21: Comparing the Sääksjärvi (1998) model and the new, leveraged model

Metric / Variable	Sääksjärvi (1998) model	New, leveraged model	Meaning and implication
Context	Strategic, software engineering based, product platform *enabled* software product development	Strategic, managerial decision-making supporting, product platform *empowered* software product development.	Changing context to better support managerial decision-making and improve its quality.
Direction of view / basis of construction	Bottom-up. Technology, product development, software as an end.	Top-down. Support the realization of business imperatives i.e. profitability, growth etc by increasing decision-making quality.	The view from the original model is turned around.
Time dimension	One-dimensional. Incorporates only pre-launch activities.	Two-dimensional. Incorporates both pre- and post-launch activities.	Adding a new time dimension fundamentally changed the nature and characteristics of the model.
View of platform	Central role. The platform is the goal, the end state and vision.	Important role. The platform facilitates development of product (family) vision.	The outcome from the Sääksjärvi model is to produce a platform. The outcome from the new model is help increase decision-making quality in platform based product development.

Metric / Variable	Sääksjärvi (1998) model	New, leveraged model	Meaning and implication
Primary target audience	Senior executives and managers with technological understanding. Also tactical level middle management.	Senior executives and managers with little or no technological understanding.	The Sääksjärvi model implies that its users have a good working knowledge of technology. The new model helps its users to understand some issues related to strategic platform based product development.
Component fits	3 balanced, harmonious, consistent, equal strength fits.	2 fits. One strong, one weak.	The Sääksjärvi model is balanced among all three components, whereas the new model is very much tilted towards one strong fit.
Motion / direction of model	Circular, continuous. No distinct beginning or end.	Straight, process-like. Clear start and finish.	Sääksjärvi's model is more conceptual, whereas the new model is more practical.

6.2 Results from study's empirical part

As for the results of the empirical part of this thesis, it is obvious that the model requires additional work and refinement. At its current stage the new model is not completely ready for a practical, real world application. This does not mean that the model, as such, could not be of use. On the contrary, this study shows that it has been of use, especially in aligning and directing decision-making on both a generic level but also at various different stages. The new model does not produce exact results but helps facilitate and coordinate in finding some possible solutions and potential outcomes for product development and its management. These issues and concerns are at best complex but the new model succeeds in providing fresh perspectives and new views to combat complexity and fight uncertainty. With the new model, managers and executives have a better chance of making informed decision, based on their likely backgrounds, without having to be experts in technology or software engineering. The new model will not prevent disasters or mistakes but should decrease their likelihood.

So what exactly is the benefit and value of the new model for Evolvis? This is difficult to answer for two reasons. Firstly, as has been previously mentioned, Evolvis has had a platform implemented well before this study took place. Therefore, one is unable to assess whether a platform would have been implemented because of the presented model or not. Secondly, a much longer time period would be needed to find out the real value of the model for Evolvis. It is therefore concluded that the new model has helped Evolvis and will continue to provide assistance and service in solving product development problems. On the other hand, the exact results, value and meaning of the new model for Evolvis are unclear, fuzzy and ambiguous.

Returning to the initial empirical research questions of whether the new model could be applied into decision-making situations in a real business context within a case company, this thesis provides an answer; yes it could. However, other questions arise from this as to the depth, relevance and meaning of this applicability, which are not so straightforward to answer, based on the information gathered for this thesis.

6.3 Practical case company recommendations

Based on this research and its results, a list of practical, implementable recommendations will be presented. These points are a way forward to help Evolvis in increasing the quality of its software product development related decisions. They are not intended as a magical list, but rather as steps that together with other organizational and managerial development initiatives will help Evolvis reach its business goals quicker and with less effort.

First, Evolvis needs to document its decision process and to align it with the overall software product development process. Without this integration continuing with a decision support model would be useless, as it would essentially act as a stand-alone application with no apparent link into the operational business of Evolvis. A second major point worth considering would be to either conduct a thorough search for a more appropriate project management model, which should contain clear elements that support decision-making along its course, or to continue with the development and refinement of the tentative, new model presented in this thesis. The researcher recommends continuing the development of the new model, because Evolvis is not in need of a new project management model. Finally, it is recommended that the personnel of Evolvis receive additional training in managing complex software related decisions and that this training be supported by computer applications.

6.4 Feasibility and applicability

The feasibility of a study refers to the degree to which something can be carried out or achieved. Alternatively, feasibility can be measured by asking if the study is reasonable enough to be believed or accepted. Applicability, on the other hand, measures to what degree the study, with its results and conclusions, can be applied either into an academic scene or into a real life environment. These concepts are examined in the following sub-chapters.

Feasibility of study

The goal of operationalisation is to produce empirically measurable equivalents for the underlying theoretical framework or model (Eskola & Suoranta 2001, 75; Hirsjärvi et al. 1997, 144). In other words, operationalisation is defined as the changing of theoretical variables into an empirically measurable form, which leads to a connection and link between theoretical concepts and reality (Soininen 1995, 73). Based on this, a conclusion must be made that scientifically justifiable operationalisation has not been achieved, as the empirical study did not produce clearly measurable variables for the new, leveraged model. The case study was used primarily as a forum for evaluation and reflection on how the results of the theoretical part of the study could be applied into a real business environment. This shortcoming considerably lowers the feasibility of this study, as it cannot be objectively assessed. It seems that on a research process level a fair level of feasibility has been achieved but on a research result level one is unable to find criteria to accept the study.

In addition, the case study was not used to calibrate the new model, as it was presented only on a conceptual level (with some tentative mathematical formulas), without actual quantitative data as a basis for its results. Therefore, based on the above and feedback received for this thesis, an average level of feasibility is believed to have been achieved, despite some of the shortcomings and problems associated with the new, leveraged model.

Applicability of study

Based on the findings, the applicability of the new model is, at best, on a generic level, and thus suited foremost as a graphically illustrated conceptual idea, rather than as a mathematical formula. Therefore, the highest level of applicability will be for corporate practitioners i.e. managers and executives, but great care must be used for the implementation and application of the new model into any new business environment. The amount of variables affecting the model's application is just too great to be able to give justifiable predictions on its success potential. Academic researchers should also be able to apply the results of the study as, and only as, a foundation for future research.

Table 22: Key feasibility and applicability issues

	Feasibility	Applicability
Theoretical part	Average. Reasonable quality of explanations. Overall believability fair.	Good as a conceptual, graphical depiction. Poor as mathematical notations → requires additional work.
Empirical part	Average. Lack of operationalisation. Only a single case analysis.	Good in case company context. Poor on generic level i.e. cannot be generalized.

6.5 Validity and reliability

Prior to examining the validity and reliability of a qualitative study, two key issues must be addressed. Firstly, separating the empirical data analysis from the actual analysis of validity is sometimes difficult, if not impossible, when compared with that of a straightforward quantitative study. Secondly, in a qualitative study it is generally easier to move among the different stages of the study - analysis, interpretation and reporting - than would be in a quantitative study. Therefore, rating a qualitative study simplifies to the examination of the actual study process and the subjectivity (and potential bias) of the researcher. It has even been argued (e.g. Wolcott 1995; Holstein & Gubrium 1995) that validity and reliability, as concepts for valuing the correctness, authenticity and accuracy of a study, do not fit into the realm of qualitative research. (Eskola & Suoranta 2001; 208, 211)

Validity of study

Kerlinger (1977, 457) defines validity with the question: "*were we measuring what we intended to measure?*" He goes on to identify three types of validity, namely content validity, criterion-related validity and construct validity (Kerlinger 1977, 457-461). Content validity refers to the representativeness or sampling adequacy of the content used in a study. In this thesis, a quantitative approach for the collection of empirical data was not used but rather a variety of qualitative information was gathered from different sources in the case company. Therefore, one could state that the content is valid in context and relation to the specific case company, as tests specifically measured what was intended to. Here, the term "*measuring*" refers to qualitatively testing the applicability of the new, leveraged model in a business environment and not on specific measurements of variables or quantitative data.

Criterion-related validity, sometimes referred to as predictive validity, seeks to answer how well the test (or tests) predicts the attribute(s) under examination (Kerlinger 1977, 459). This type of validity analysis poses several difficulties, as no explicit tests to predict changes or fluctuations in any attributes have been performed. Consequently, one must conclude that, for this study the use of criterion-based validity is not relevant. Finally, construct validity is used when questions arise in relation to the factors that explain the variance of the test, e.g. what else do the results show or factors explain (Kerlinger 1977, 461). Again, it is obvious that construct validity is of no help in analyzing the overall validity of this study.

Where Kerlinger's discussions on validity center on quantitative research, Hammersley (1993) focuses his views on validity in qualitative, social studies. According to Hammersley, validity can be assessed based on a three-step model. First, one asks if the proposed statement or theorem could be accepted without further proof or reasoning. This is done because the proposition might be self-evident. Secondly, one must be sure of the credibility of the proposed statements, which can be evaluated by assessing e.g. the research circumstances, used methods and presented definitions. Finally, one asks about the relevance of the study, by questioning the generic use and practical meaning of the research.

Answering the points put forth by Hammersley, the propositions stated herein, both in the theoretical and empirical parts, cannot be accepted without further proof and additional research. Although logical and structured sets of reasoning and explanations have been presented to support this study, the presented findings remain tentative and can act only as a basis for further research. As for the research process and methods used, it is argued that because of the simplicity of the empirical research process the only relevant question is about the potential bias of the researcher and the possible non-repeatability of the empirical study. Finally, this study only has relevance on a conceptual theoretical level. The findings from the empirical research cannot be generalized and thus have little or no relevance for anyone except the management and employees of Evolvis.

Because of the very (qualitative) nature of this study, it is not possible to adequately answer questions relating to the three types of validity presented by Kerlinger. However, Hammersley's three-step validity analysis for qualitative research has been covered. The only remaining question is the original question defining validity, namely measuring what was intended to measure. The initial objective for this study was to produce a qualitative case company analysis for the empirical part. As such, this has been completed, to the best of the researcher's knowledge and capabilities.

Reliability of study

Comparing the results obtained by a different researcher, who carried out exactly the same research, can test the reliability of a study. In other words, a new researcher should obtain the same results and findings and draw the same conclusions as the initial researcher, if she follows exactly the same procedures as the initial researcher, in conducting the same study all over again. Boudreau et al. (2001, 5) state that reliability is about measurement accuracy i.e. the extent to which consistent and error-free results are produced. Yin (2003, 37) continues along the same lines in stating that "*the goal of reliability is to minimize the errors and biases in a study*". Therefore, the reliability of a study is measured by its ability to produce non-random results. Moreover, good documentation enhances reliability because it allows for the study to be repeated at a later date. (Yin 2003, 37-39; Hirsjärvi et al. 1997, 213)

In this thesis, reliability must be evaluated on three different levels. Firstly, by asking: "*could the new, leveraged model have been produced by a different researcher?*" Obviously, the answer to this is yes, as the new model is not based on anything fundamentally new, but rather on recycling used concepts and presenting them in a fresh perspective and context.

The second question evaluating reliability could be written as: "*is the empirical data collection and analysis reliable?*" In terms of gathering data, answers from the case company's employees should be independent of the researcher, i.e. similar answers should be obtained regardless of who is asking the questions. Here a substantial loss of study credibility and reliability is witnessed, as the answer to this question is negative. The close personal ties and long employment history of the researcher in Evolvis are key issues affecting the outcome and results of this thesis. It would be hard, if not impossible, to foresee a situation, where an external researcher could, on the one hand, receive all of the same data and, on the other hand, produce the same results and interpretations.

Finally, the reliability of this study can be examined by assessing the validity of the conclusions made by the researcher. Alasuutari (1995, 132) argues that the level of logical result and data conclusion explanation can be used to assess the reliability of a study. Based on this, previous research to either support or contradict the conclusions made here should have been found. However, as no such research was found, it seems to indicate either a limited effort in searching previous studies or the fact that they simply do not exist. In light of this statement, it must be mentioned and pointed out that the objective of this study was never to find contradicting or supporting research, but rather to participate in a continuing evolution on the combined discourses of product platform based software product development and managerial decision-making. Table 23 summarizes the key points on validity and reliability.

Table 23: Key validity and reliability issues

	Validity	Reliability
Theoretical part	Not available. Nothing to measure. New, leveraged model requires additional proof and explanations. Problems with validating the correctness of the mathematical formulas and equations.	Good. New, leveraged model could be produced by anyone. Problems with the reliability of the mathematical formulas and equations.
Empirical part	Good. "*Measurement*" of what was intended i.e. preliminary, tentative tests on theoretical model applicability.	Poor. Empirical part is very tightly linked to the researcher → potential bias.

6.6 Key issues and concerns

The last two sub-chapters have examined this thesis based on its feasibility, applicability, validity and reliability. Continuing in the same direction, this analysis and evaluation will be deepened by discussing some of the other key issues and concerns associated with both the theoretical model and empirical research of this thesis. The following discussions and listings are by no means comprehensive, but rather offer a direction setting stage for more in-depth discussions concerning this thesis.

6.6.1 Theoretical model

There is no such thing as perfect errorless, scientific research. Just as every research has, either major or minor, flaws and mistakes so too does this one. Below is a list of some of the major irrationalities, discrepancies and other concerns relating to the in-experience, bounded rationality and potential bias of the researcher. The points are presented in no special order.

Scientific credibility of material used. Some seminal material used herein does not conform to the strictest principles and guidelines of scientific research. E.g. the Sääksjärvi (1998) product platform model is published as a TEKES technology report in the Finnish language, meaning that it has not been subject to the evaluation and scrutiny of a larger, international academic audience.

Another example is "*Information Rules*" which was used as the primary source of information for leveraging the technology strategy component. These facts decrease the academic value and dilute the scientific credibility of this thesis.

Seemingly unrelated concepts, ideas and models used in the leveraging of the new model. Firstly, the model presented can be seen as very fragmented, being made up of small parts and even sometimes incompatible concepts, most of which do, however, originate directly from or link to the realm of information systems science. The model can also leave some readers with the impression of being highly abstract and theoretical with no apparent link to the real world. Finally, the mathematical notations will not appeal to all.

Focus on individual components of the new model. It is true that the focus was very much on the individual components of the Sääksjärvi (1998) model, because this eased the analysis considerably by breaking up a larger model into more easily handled smaller parts. However, the model as a complete, holistic entity was also examined.

Use of other concepts to leverage the Sääksjärvi (1998) model. The presented concepts used for leveraging the Sääksjärvi (1998) model are not the only ones possible. Portfolio theory is an example of a concept that could have also been used for leveraging the initial model. Managing the components (i.e. the assets) of the Sääksjärvi (1998) model, based on expected risk and return potential, would have resulting in a portfolio theory application. A second example of a concept that could have been used is game theory, which could have calculated dominant strategies for each component of the Sääksjärvi (1998) model, based on resources used, information of the other components and expected component strategies. In this way, the components would be seen as either competing for the same resources (i.e. an oligopoly) or colluding for their overall benefit (i.e. a cartel).

On selecting the Sääksjärvi (1998) model. This thesis has selected the Sääksjärvi (1998) model without a thorough analysis of other possible platform models, decreasing its scientific rigor. In addition, this thesis has failed to answer a more fundamental question, namely what development model would produce the best products and what decision criteria would it need.

On the mathematical formulas and equations. It is essentially highly questionable if the presented mathematical formulas and equations provide any real increase in understanding of the new, leveraged model. They are foremost just another form of presentation, not a set of tools for adding comprehension.

Use of the term "new, leveraged model". Is the presented model really new? Is it even a leveraged model? Isn't it just a compilation of seemingly unrelated ideas and concepts packaged and presented in a fancy graphical layout? All are relevant questions, and can only be answered by the reader or examiner. However, it is stated that the use of the term "*new, leveraged model*" has only been used to distinguish it from the original Sääksjärvi (1998) model and, as such, does not imply any specific level or relation of newness or correctness. On the other hand, the initial Sääksjärvi (1998) model has been extended to such an extent that it is difficult, if not impossible, to distinguish it from the new, leveraged model because of their differing assumptions and postulations. Therefore, it could be argued that a more appropriate term, simply "*model*", should be used. This has, however, not been done because of the need to stress the importance of the Sääksjärvi (1998) model in designing and producing the new, leveraged model. Ultimately it is Sääksjärvi who has done the majority of work in facilitating the design and production of the new, leveraged model.

6.6.2 Empirical research

This chapter ends with some critical insights into the problems and weaknesses in the empirical part of this study. Again, the following list is not comprehensive, but is provided as a basis for further discussions about the merits and shortcomings of this thesis.

Choice of case company. BNL is a communications and PR company and has thus very little to do with software product development. Although Evolvis incorporates a fair amount of software product development to produce its services, it cannot be categorized as a pure software company. Choosing a real software product development company would have been more appropriate for this study, but would have been impossible to implement, due to logistical and time constraints. Secondly, it is believed that the scale and scope of software product development in Evolvis was sufficient enough to carry out the empirical research. In part, the choice of Evolvis for a case company analysis can be said to be a convenience sample.

Researcher bias. The researcher is potentially biased with his long history within the case company and close ties to its personnel. However, it is believed that using appropriate methods and analysis processes, e.g. a formal interview outline, has minimized the effect of potential bias. In case analyses, with which the researcher has a close and personal relation, it is nearly impossible to prevent or eliminate possible bias. Often resulting in subjective interpretations or a lack of openness with regard to possible contrary findings, researcher bias cannot be completely removed from qualitative studies.

Scientific rigor vs. relevance. Applying equally to both the theoretical and empirical parts of thesis, one could question the level of scientific rigor used herein. It is obvious throughout this study that only an average level of rigor is achieved, due to the preliminary nature and general infancy of the new, leveraged model and the decision to conduct a qualitative case analysis. However, an above average level of relevance is reached from the perspective of future researchers wanting to pursue product platform related ideas and issues. In addition, the qualitative case study should allow other managers and executives to use and apply some of the points gained from the Evolvis analysis.

Analysis vs. interpretations. Isn't the empirical case study just an interpretation of the researcher's perceived reality and long history as an Evolvis employee? Isn't the so-called "*analysis*" just a collection of subjective conclusions that cannot be objectively validated? Responding to these relevant questions, it is stated that the empirical part of the study was conducted in two distinct parts. First, the information was collected and then analyzed, next it was interpreted. It is only natural that the interpretation is subjective and closely related with and associated to the researcher. This fact has never been hidden and has been discussed openly throughout the study.

A qualitative methodology? Despite the existence of great heterogeneity within qualitative methodology literature, this thesis has tried to adhere to the three fundamental assumptions associated with qualitative methods: a holistic new, an inductive approach, and a naturalistic inquiry (Rudestam & Newton 2001, 36-37). Of these a holistic approach underlines that the whole is different from the sum of its parts. A holistic view seeks to understand phenomena in their entirety, as has been the situation with the case company analysis, in order to develop a thorough and complete understanding of the issues at hand. Secondly, an inductive approach begins with specific observations and then moves on toward identifying general patterns or developing generic theories. This inductive style is witnessed in the continuum from the preliminary applicability testing of the Sääksjärvi (1998) model to the application of the new, leveraged model. Finally, a naturalistic inquiry intends to understand phenomena in their natural environments, which is evident from trying to apply and examine the new, leveraged model in a real business context.

7. SUMMARY AND CONCLUSIONS

This thesis ends with a brief summary and some closing remarks. The questions being addressed in this section include: What did the study provide? How did the study help in solving the initial problem? What are the conclusions and their consequences and implications?

Developing commercially successful software products is an inherently uncertain activity, evident from a vast myriad of prior studies and research. Even the logical, rational and quantitative methods and models generally used throughout the IS domain have not been able to produce lasting solutions for the issues discussed herein. Maybe the answer lies at a crossroads linking the soft, irrational, illogical and qualitative aspects of human psychology to the hard rigor of information systems science.

However, uncertainty, risks and failures are real IS world phenomena, witnessed everyday. These risks of uncertainty are often materialized by non-technology oriented executives and managers making hasty and unjustifiable decisions, leading to schedule and cost overruns and unaligned, uncoordinated activities, all of which could be reasons for eluding market success. Clearly there is an articulated need for helping these managers and executives in increasing the quality level of their software development related decisions.

As this study has shown, it seems fair to state that product platform based software product development is even more difficult, by adding several components and dimensions to the overall puzzle of developing commercially successful software products. Again, managers and executives have to make, at best, complex decisions with various alternatives, differing combinations and far reaching options. However, all is not lost, as this thesis has tried to design, produce and apply a new decision-supporting model to help managers and executives to increase the quality of their decisions by introducing simple logic, intuitive methods and a fresh perspective to guide and direct the identification and solving of software product development related issues and concerns.

In this study one has examined product platforms, not only as a possible method for developing software products, but foremost as a basis for increasing the quality of managerial decision-making regarding software product development. If designed, implemented and maintained correctly, a product platform enabled and empowered decision-support model can also provide strategic guidance by pointing a vector of direction to an overall business vision. At its highest level of abstraction, product platform thinking is a philosophy steering its implementer towards increased strategic alignment and integration between business objectives and software product development.

This study did not produce final or absolute answers to the research problems and their associated research questions, partly because of the ambiguity of the initial problem and partly because of the experimentational nature of the research project. Rather, the objective was to contribute some new ideas to the ongoing research and discussions on product platforms and software product development. The presented model is one possible approach to the problems addressed; it is not the only one or even perhaps a viable one.

Finally, the question of what the study provided, remains. The contributions of the thesis are twofold. The design and production of a new, tentative product platform enabled and empowered decision support model is the primary asset derivable from herein. Second, and based on the previous asset, this study has been able to provide the case company with some practical recommendations that should help on a strategic guidance tier but also on a more everyday operational level. The real impact from these recommendations remains to be seen.

7.1 Key findings and lessons learned

The key findings from this thesis can be categorized on two levels. On a theoretical level, the key finding is that the presentation of a new, leveraged model, based on the Sääksjärvi (1998) model, to support managerial decision-making in product platform based software product development provided mixed and ambiguous results. On the one hand it succeeded in certain aspects, whereas failed in others, especially the formulation and presentation of the 3D spatial model component. The new model's main theoretical contribution is to act as a basis for further research. The validity and reliability of the new model and its components is yet to be proven.

On an empirical level the key finding is not the generation of a set of recommendations for the case company, but rather the creation of new, explicit and implicit, organizational knowledge about the existence of the company, its strategy and vision, its products and services, its processes and critical success factors, its employees and their relations etc. This finding is perhaps unexpected and surprising as the list of recommendations has potential and a good chance of providing increased profitability to Evolvis if executed and implemented properly. In a small organization conducting a study of this magnitude has forced everyone to rethink, re-evaluate, rearticulate and reproduce all of their organizational information and knowledge. Perhaps this process of socialization has reinforced and strengthened the positive and successful attributes of Evolvis's business while identifying and possibly even correcting some of the unsuccessful ones. Herein lies the true value of this study for Evolvis.

Lessons learned

According to Järvinen & Järvinen (2000, 33) a new proposed theory should fulfill at least two conditions. First it must accurately and fairly represent and depict the reality it tries to capture. Second it should be better than previous theories and this should be proven. The first condition is usually proven with an empirical study, while the second is addressed by comparing the new theory to its best rival. In this thesis, the empirical representation and depiction of the new, leveraged model has been attempted in a case company environment. The sample of a single company is not sufficient to allow a broader generalization and induction. However, the results are applicable to the specific case company. As for the second condition of comparing the theoretical model to its best rival, it must be stated that the new, leveraged model does not have one. Although the Sääksjärvi (1998) model and the new, leveraged model are compared to each other in chapter 6, this does not do either model complete justice as their underlying assumptions differ considerably. Table 24 presents some of the academic lessons learned from this study and then suggests some possible corrections.

Table 24: Lessons learned and some correction suggestions

Lessons learned	Comments and correction suggestions
Problems with broad scope of research. Incorporating decision support with product platforms proved to be too much for one study.	Successful research is based on clearly articulating and defining the research, its design and strategy, by imposing restrictions and limitations. A good study is always founded on answering a niche problem.
Conducting a successful empirical case study is not easy and by no means uncomplicated.	When compared with the statistical concepts in quantitative research a case study may seem simple and straightforward. This is an illusion.
Reporting a thesis in a brief and concise written format requires additional training.	Writing a long thesis is much easier than writing a short one. This thesis is too long; its abbreviation proved to be too great a task for the inexperienced researcher.

7.2 Future research suggestions

It seems that with this thesis more questions have been produced than answers. Partially this was expected, as the main contribution is clearly the participation in an ongoing discourse rather than sound, justifiable and valid results.

During the research and writing of this thesis, many interesting ideas, concepts and frameworks have been come across. These could have been added and incorporated to either the new model or its business environment application. However, this has been left for future researchers. Table 25 presents some practical suggestions for further research.

Table 25: Suggestions for future research

Future research	Description
Model refinement	The presented model needs a considerable amount of additional work to be a viable and reliable entity in managerial decision support. The model requires further work and refinement in e.g. the theoretical foundations of the model and its components and the rigor of its mathematical notations.
Model computerization	As presented earlier, the model works reasonably well in a graphical format. However, this is not adequate to satisfy the needs of most managers and executives. A formal and neutral computing environment is needed to justify wider use of the model. Perhaps an Excel add-in module or simple web application could be considered.
Quantitative data	The model requires quantitative numerical data to facilitate operationalisation and computation attempts. Without this the model will remain on a graphical, pictured level, without being able to produce actual quantitative, comparable results.
Improved qualitative case analyses	The validity and reliability of the results from this thesis could be either validated or denounced based on further and improved qualitative case analyses, preferably with several new, pure software product companies.

REFERENCES

Literature (*Please note that literature marked with a star, *, is not directly cited in the text*)

Abrahamsson, P. & Salo, O. & Ronkainen, J. & Warsta, J. (2002). *Agile Software Development Methods – Review and Analysis*. VTT Publications 478.

Ahituv, N. & Neumann, S. (1987). *Decision Making and the Value of Information*. An article part of a reading package for the Fall 2002 course "Critical Issues in Information Systems Research", 37E440, lectured by prof. Matti Rossi.

Alasuutari, P. (1995). *Laadullinen tutkimus*. 3. edition. Vastapaino Oy, Tampere

* Applegate, L. (1994). *Managing in an Information Age: Transforming the Organization for the 1990's*. An article part of a reading package for the Fall 2002 course "Critical Issues in Information Systems Research", 37E440, lectured by prof. Matti Rossi.

Baird, B.F. (1989). *Managerial Decisions Under Uncertainty. An introduction to the analysis of decision-making*. John Wiley & Sons, Inc

Baldwin, C.Y. & Clark, K.B. (2000). *Design Rules – The Power of Modularity*. The MIT Press

Beck, K. (2000). *Extreme Programming Explained – Embrace Change*. Addison Wesley

* Benaroch, M. & Kauffman, R.J. (1999). *A Case for Using Real Options Pricing Analysis to Evaluate Information Technology Project Investments*. Information Systems Research March 1999, vol.10. number 1.

* Benita, S. & Tolkowsky, E. (2002). *Real Options – an Introduction and an Application to R&D Valuation*. The Engineering Economist 2002, vol. 47, number 2.

Bodie, Z. & Merton, R.C. (2000). *Finance*. Prentice Hall.

Boudreau, M-C. & Gefen, D. & Straub, D.W. (2001). *Validation in Information Systems Research: A State-of-the-Art Assessment*. MIS Quarterly Vol. 25 No. 1 pp. 1-16 / March 2001

Brealey, R.A. & Myers, S.C. (2000). *Principles of Corporate Finance*. 6. edition. Irwin McGraw-Hill.

* Christensen, C.M. (1997). *The Innovator's dilemma – When New Technologies Cause Great Firms to Fail*. Harvard Business School Press.

Chu, P.C. & Spires, E.E. (2000). *The Joint Effects of Effort and Quality on Decision Strategy Choice with Computerized Decision Aids*. Decision Sciences, vol. 31, no. 2, winter 2000.

Cockburn, A. (2001). *Agile Software Development*. Addison-Wesley.

Coghlan, D. & Brannick, T. (2001). *Doing Action Research in Your Own Organization*. Sage.

Copeland, T. & Antikarov, V. (2001). *Real Options – A Practitioner's Guide*. Texere.

* Cottrell, T. & Sick, G. (2002). *Real Options and Follower Strategies: the Loss of Real Option Value to First-Mover Advantage*. The Engineering Economist 2002, vol. 47, number 3.

* Davis, C.R. (2002). *Calculated Risk: a Framework for Evaluating Product Development*. MIT Sloan Management Review, summer 2002.

* D'Souza, F. (2002). *Putting Real Options to Work to Improve Project Planning*. Harvard Management Update, August 2002, vol. 7, issue8.

Earl, M. (1989). *Management Strategies for Information Technology*. Prentice-Hall International Limited.

* Edwards, J.S. & Duan, Y. & Robins, P.C. (2000). *An Analysis of Expert Systems for Business Decision-making at Different Levels and in Different Roles*. European Journal of Information Systems, vol. 9, pp 36-46.

* Elton, E.J. & Gruber, M.J. (1995). *Modern Portfolio Theory and Investment Analysis*. 5. edition. John Wiley & Sons, Inc.

Erdogmus, H. (2002). *Valuation of Learning Options in Software Development under Private and Market Risk*. The Engineering Economist 2002, vol. 47, number 3.

Eskola, J. & Suoranta, J. (2001). *Johdatus laadulliseen tutkimukseen*. 5. edition. Vastapaino Oy, Tampere.

* Galliers, R. & Baker, B. (1994). *Strategic Information Management*. Butterworth-Heinemann

Gawer, A & Cusumano, M.A. (2002). *Platform Leadership – How Intel, Microsoft, and Cisco Drive Industry Innovation*. Harvard Business School Press.

Gibbs, W.W. (1994). *Software's Chronic Crisis*. Scientific American, September 1994, pages 86-95.

Hammersley, M. (1993). *Social Research – Philosophy, Politics and Practice*. Sage.

Hess, T.J. & Rees, L.P. & Rakes, T.R. (2000). *Using Autonomous Software Agents to Create the Next Generation of Decision Support Systems*. Decision Sciences, vol. 31, no. 1, winter 2002.

Hirsjärvi, S. & Remes, P. & Sajavaara, P. (1997). *Tutki ja kirjoita*. 6. edition. Kustannusosakeyhtiö Tammi, Helsinki.

Holstein, J.A. & Gubrium, J.F. (1995). *The Active Interview*. Sage.

* Hunton, J.E. (1996). *Involving Information System Users in Defining System Requirements: The Influence of Procedural Justice Perceptions on User Attitudes and Performance*. Decision Sciences, vol. 27, no. 4, fall 1996.

Jehle, G.A. & Reny, P.J. (2001). *Advanced Microeconomic Theory*. 2. edition. Addison Wesley Longman.

Jones, C. (1994). *Assessment and Control of Software Risks*. Yourdon Press.

Järvinen, P. & Järvinen, A. (2000). *Tutkimustyön metodeista*. 2. edition. Tampereen Yliopistopaino Oy.

Kangasharju, H. & Majapuro, M. (1999). *Tutkimusraportin kirjoittaminen*. Helsingin Kauppakorkeakoulun julkaisuja.

Kerlinger, F.N. (1977). *Foundations of Behavioral Research*. Holt, Rinehart and Winston Inc.

Kim, J. & Lee, J. & Han, K. & Lee, M. (2002). *Businesses as Buildings: Metrics for the Architectural Quality of Internet Businesses*. Information Systems Research vol.13, no. 3, September 2002, pp. 239-254.

* Land, F.F. & Kennedy-McGregor, M. (1987). *Information and Information Systems: Concepts and Perspectives.* An article part of a reading package for the Fall 2002 course "Critical Issues in Information Systems Research", 37E440, lectured by prof. Matti Rossi.

Landsburg, S.E. (1999). *Price Theory & Applications*. 4. edition. International Thomson Publishing.

Lederer, A.L. & Prasad, J. (1992). *Nine Management Guidelines for Better Cost Estimating.* Communications of the ACM, February 1992, pages 51-59.

Lehman, M.M. (1998). *Software's Future: Managing Evolution.* IEEE Software. Volume 15, Issue 1.

McConnell, S. (1996). *Rapid Development – Taming Wild Software Schedules*. Microsoft Press.

McGrath, M.E. (1995). *Product Strategy for High-Technology Companies*. Irvin.

Meyer, M.H. & Lehnerd, A.P. (1997). *The Power of Product Platforms – Building Value and Cost Leadership*. The Free Press.

Meyer, M.H. & Seliger, R. (1998). *Product Platforms in Software Development*. Sloan Management Review. Fall 1998.

Meyer, M.H., Terzakian, P. & Utterback, J.M. (1997). *Metrics for Managing Research and Development in the Context of the Product Family*. Management Science, vol. 43, no. 1, January 1997, pages 88-111.

Meyer, M.H. & Utterback, J.M. (1993). *The Product Family and the Dynamics of Core Capability*. Sloan Management Review, spring 1993, pages 29-47.

Nandhakumar, J. & Avison, J. (1999). *The Fiction of Methodological Development: A Field Study of Information Systems Development*. Information Technology & People 12(2): 176-191.

Niskanen, J. & Niskanen, M. (2000.) *Yritysrahoitus*. Oy Edita Ab.

* Parker, M. (1996). *Strategic Transformation and Information Technology*. Prentice-Hall International Limited.

* Pelkonen, T. et al. (2003). *Digimedia 2003 – Nykytila ja Kasvunäkymiä*. Kauppa- ja Teollisuusministeriö, LTT-Tutkimus Oy.

Perloff, J.M. (2001). *Microeconomics*. 2. edition. Addison Wesley.

Pine, B. (1993). *Mass Customization – The New Frontier in Business Competition*. Harvard Business School Press.

Rajala, R. (2000). *Applying Product Platform Thinking to the Development of Internet Application Frameworks*. Master's thesis, Helsinki School of Economics.

Ranganathan, C. & Sethi, V. (2002). *Rationality in Strategic Information Technology Decisions: The Impact of Shared Domain Knowledge and IT Unit Structure*. Decision Sciences, vol. 33, no. 1, winter 2002.

* Rudestam, K.E. & Newton, R.R. (2001). *Surviving Your Dissertation – A Comprehensive Guide to Content and Process*. 2. edition. Sage.

* Sambamurthy, V. & Kirsch, L.J. (2000). *An Integrative Framework of the Information Systems Development Process*. Decision Sciences, vol. 31, no. 2, spring 2000.

Schmidt, J.B & Montoya-Weiss, M.M. & Massey, A.P. (2001). *New Product Development Decision-making Effectiveness: Comparing Individuals, Face-to-face, and Virtual Teams*. Decision Sciences, vol. 32, no. 4, fall 2001.

Shapiro, C. & Varian, H. (1999). *Information Rules – A Strategic Guide to the Network Economy*. Harvard Business School Press.

* Simon, C. & Blume, L. (1994). *Mathematics for Economists*. W. W. Norton & Company, Inc.

Singh, M.G. (1994). *Decision Technologies for Supporting the Interplay Between Qualitative and Quantitative Aspects of Managerial Decision-making*. Mathematics and Computers in Simulation, vol. 36, no. 2. June 1994, pp. 103-114.

Soininen, M. (1995). *Tieteellisen tutkimuksen perusteet*. Turun yliopiston täydennyskoulutuskeskuksen julkaisuja A:43.

Sääksjärvi, M. (1998). *Tuoterunko – uusi ajattelu ohjelmistotuotteiden strategisessa kehittämisessä*. Teknologiakatsaus 62/98, TEKES

Sääksjärvi, M. (2002). *Software Application Platforms: From Product Architecture to Integrated Application Strategy*. Working papers W-309, Helsinki School of Economics.

* Tan, K.C. (2001). *A Structural Equation Model of New Product Design and Development*. Decision Sciences, vol. 32, no. 2, spring 2001.

Teng, J.T.C. & Calhoun, K.J. (1996). *Organizational Computing as a Facilitator of Operational and Managerial Decision-making: An Exploratory Study of Managers' Perceptions*. Decision Sciences, vol. 27, no. 4, fall 1996.

Truex, D.P. & Baskerville, R. & Travis, J. (2000). *Amethodical Systems Development: The deferred Meaning of Systems Development Methods*. Accounting, Management and Information Technology 10: 53-79.

Turban, E. & Aronson, J. (1998). *Decision Support Systems and Intelligent Systems*. 5. edition. Prentice-Hall International Limited.

Ullrich, K.T. & Eppinger, S.D. (1995). *Product Design and Development*. McGraw Hill.

Uusitalo, H. (1991). *Tiede, Tutkimus ja Tutkielma – Johdatus tutkielman maailmaan*. Werner Söderström Osakeyhtiö.

Van Bruggen, G. & Wierenga, B. (2001). *Matching Management Support Systems and Managerial Problem-solving Modes: The Key to Effective Decision Support*. European Management Journal vol. 19, no. 3, pp. 228-238.

Venkatesh, V. & Speier, C. & Morris, M.G. (2002). *User Acceptance Enablers in Individual Decision-making About Technology: Toward an Integrated Model*. Decision Sciences, vol. 33, no. 2, spring 2002.

Verma, R. & Thompson, G.M. & Moore, W.L. & Louviere J.J. (2001). *Effective Design of Products / Services: An Approach Based on Integration of Marketing and Operations Management Decisions*. Decision Sciences, vol. 32, no. 1, winter 2001.

Weill, P. & Broadbent, M. (1998). *Leveraging the New Infrastructure – How Market Leaders Capitalize on Information Technology*. Harvard Business School Press.

Welke, R.J. (1994). *The Shifting Software Development Paradigm*. Database vol. 25, no.4, November 1994, 9-16.

Wheelwright, S.C. & Clark, K.B. (1992). *Revolutionizing New Product Development*. The Free Press.

Wolcott, H.F. (1995). *The Art of Fieldwork.* AltaMira.

Yin, R. (2003). *Case Study Research – Design and Methods*. 3. edition. Sage

Internet addresses

http://www.bnl.fi

http://www.cmswatch.com

http://www.evolvis.net

http://www.extremeprogramming.org

http://www.midgard-project.org

http://www.nadminstudio.com

Other material

BNL company presentation slides (23.09.2002). Available upon request.

Evolvis business plan, versions 1.0 – 1.4 (February 2001 – April 2001). Available upon request, subject to confidentiality.

Evolvis product development memos (19.11.2001, 04.08.2002). Available upon request, subject to confidentiality.

Krishnan, V.V. (2002). *Leveraged Innovation in Mobile Markets: Platform-based Products and Services*. Presentation slides. Available upon request.

APPENDIX A: Interview themes and document outline

All of the official interviews were conducted based around an open structure evolving around a loose set of themes as opposed to a tightly constrained and formally articulated questionnaire (see e.g. Eskola & Suoranta 2001, 86-88 for elaboration on different interview styles). As the interview process was conducted on an informal basis, resembling a conversation or dialogue, every theme was not examined with every interviewee. The interviews concentrated on themes most relevant for each individual.

The interviewee

The aim is to gather data and information relating to the background of the individual, his or her level of education or other specific formal or informal qualifications. Finally a brief account of the interviewee's work history is needed. Themes:

- Background, education, work history etc.

Current work

Acquire information about the role and responsibilities of the individual in Evolvis. Have they changed or developed over the course of employment? Themes:

- Role in Evolvis, current responsibilities, possible change in either role or responsibilities etc.

Evolvis, organizational issues

Collect information on the organizational issues and aspects relating to Evolvis. How is Evolvis organized? What is the culture and philosophy like? Themes:

- Organization, culture, philosophy, procedures, daily operations etc.

Evolvis, decision-making and processes

The goal is to understand and model the processes and decision-making of Evolvis. How do they integrate with one another? Themes:

- Organization processes, decision-making etc.

Evolvis, product and platform development

Gather information relating to product and platform development. Areas of special interest include current platform implementation and integration with applications and EAS in general. What is the product family vision of Evolvis? Themes:

- Product platform, product development, EAS and applications, product family vision etc.

APPENDIX B: Interviewees and interview schedule

The following people, all current employees of Evolvis Euro RSCG, have participated in the case study. All have contributed comments, views, opinions and facts for the empirical portion of this thesis. The list, presented in table 26, is arranged alphabetically, according to surname. Due to confidentiality and privacy issues, only the initials have been used as identification.

Table 26: Interviewees and interview schedule

Initials	Position	Interview date(s)
JH	Consultant	18.08.2003
KK	Group director	23.05., 01.07.2003
MK	Graphic designer	08.08.2003
JL	Graphic designer	08.08.2003
MN	Programmer	18.08.2003
KR	Communications consultant	08.08.2003
AR	Graphic designer	08.08.2003
LT	Software engineer	05.08.2003
AT	Systems engineer	24.06.2003
KYA	Programmer	05.08.2003

APPENDIX C: Preliminary Sääksjärvi model applicability test

As discussed in sub-chapters 3.4.1 and 3.4.2 of this thesis, the preliminary applicability test of the Sääksjärvi (1998) model was divided into three distinct parts. The test started with an initial assessment of the complete, holistic model with three general questions. The goal with these questions was to position the model in accordance with the strategic-tactical-operational continuum, as expressed in chapter 2.2. The questions[34], for part one, were:

Part I: General questions		
Number	*Question*	*Comments*[35]
Q1	On a strategic level, the Sääksjärvi model is suited for leading software product development.	These questions map Sääksjärvi's model onto a strategic-tactical-operational grid, which is used to evaluate the most appropriate corporate level for the model's usage.
Q2	On a tactical level, the Sääksjärvi model is suited for managing software product development.	
Q3	On an operational level, the Sääksjärvi model is suited for implementing software product development.	

The second part of the questionnaire moved on to analyze the individual components of Sääksjärvi's (1998) model. Each component was evaluated based on the characterization found in sub-chapter 1.5.3 i.e. structural firmness, functional convenience and representational delight. Therefore, part two of the questionnaire contained the following questions:

Part II: Component questions		
Number	*Question*	*Comments*
Q4a	Of the three components in Sääksjärvi's model, product architecture has a good level of structural firmness.	These questions map the components of Sääksjärvi's model with Kim et al.'s (2002) characterization.
Q4b	Of the three components in Sääksjärvi's model, product architecture has a good level of functional convenience.	
Q4c	Of the three components in Sääksjärvi's model, product architecture has a good level of representational delight.	
Q5a	Of the three components in Sääksjärvi's model, future extensions has a good level of structural firmness.	
Q5b	Of the three components in Sääksjärvi's model, future extensions has a good level of functional convenience.	
Q5c	Of the three components in Sääksjärvi's model, future extensions has a good level of representational delight.	
Q6a	Of the three components in Sääksjärvi's model, technology strategy has a good level of structural firmness.	

[34] Originally the questions were in Finnish. Here, they have been translated into English.

[35] These comments did not appear on the original question sheet.

Part II: Component questions		
Q6b	Of the three components in Sääksjärvi's model, technology strategy has a good level of functional convenience.	
Q6c	Of the three components in Sääksjärvi's model, technology strategy has a good level of representational delight.	

Finally, the third part of the preliminary test provided the chance to answer open-ended questions, considerably increasing the flexibility and value of the examination. The questions were designed to be informal and non-restricting to allow for a flow of new, fresh ideas for the development of Sääksjärvi's (1998) model and its components. The questions were:

Part III: Open-ended questions		
Number	*Question*	*Comments*
Q7	How could the Sääksjärvi model, on a holistic level, be developed?	The intention was to gather overall information about the possible evolution and development of the model.
Q8	How could the components of Sääksjärvi's model be developed?	The aim was to acquire information on the improvement of the individual components.
Q9	How does (or could) the Sääksjärvi model help increase the effectiveness and quality of software product development decisions in Evolvis?	With this question the plan was to link Sääksjärvi's model to the operational decision-making in Evolvis.
Q10	Other comments and feedback for the researcher.	---